aws

Entwicklerhandbuch

Amazon DocumentDB

Amazon DocumentDB: Entwicklerhandbuch

Table of Contents

Was ist Amazon DocumentDB (mit MongoDB-Kompatibilität)

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein schneller, zuverlässiger und vollständig verwalteter Datenbankservice. Amazon DocumentDB erleichtert das Einrichten, Betreiben und Skalieren von MongoDB-kompatiblen Datenbanken in der Cloud. Mit Amazon DocumentDB können Sie denselben Anwendungscode ausführen und dieselben Treiber und Tools verwenden, die Sie mit MongoDB verwenden.

Bevor Sie Amazon DocumentDB verwenden, sollten Sie sich die unter beschriebenen Konzepte und Funktionen ansehenSo funktioniert's. Anschließend führen Sie die Schritte unter Handbuch „Erste Schritte“ aus.

Themen

Übersicht über Amazon DocumentDB

Im Folgenden sind einige allgemeine Funktionen von Amazon DocumentDB aufgeführt:

- Amazon DocumentDB unterstützt zwei Arten von Clustern: Instance-basierte Cluster und elastische Cluster. Elastic Cluster unterstützen Workloads mit Millionen Lese-/Schreibvorgängen pro Sekunde und Petabyte Speicherkapazität. Weitere Informationen zu elastischen Clustern finden Sie unter Verwenden von elastischen Amazon DocumentDB-Clustern. Der folgende Inhalt bezieht sich auf Instance-basierte AmazonAmazon DocumentDB-Cluster.

- Amazon DocumentDB vergrößert automatisch die Größe Ihres Speichervolumens, wenn Ihr Datenbankspeicherbedarf wächst. Ihr Speichervolumen wächst in Schritten von 10 GB bis zu einem Maximum von 128 TiB. Sie müssen in Hinblick auf zukünftiges Wachstum keinen zusätzlichen Speicher für Ihren Cluster bereitstellen.
- Mit Amazon DocumentDB können Sie den Lesedurchsatz erhöhen, um Anwendungsanfragen mit hohem Volumen zu unterstützen, indem Sie bis zu 15 Replikat-Instances erstellen. Amazon DocumentDB-Replikate nutzen denselben zugrunde liegenden Speicher, senken die Kosten und vermeiden Schreibvorgänge auf den Replikatknoten. Diese Funktion gibt mehr Rechenleistung frei, um Leseanforderungen zu bearbeiten, und reduziert die Verzögerungszeit des Replikats – oft bis zu einstelligen Millisekunden. Sie können Replikate unabhängig von der Größe des Speichervolumens innerhalb von Minuten hinzufügen. Amazon DocumentDB bietet auch einen Reader-Endpunkt, sodass die Anwendung eine Verbindung herstellen kann, ohne Replikate verfolgen zu müssen, wenn sie hinzugefügt und entfernt werden.
- Mit Amazon DocumentDB können Sie die Rechen- und Speicherressourcen für jede Ihrer Instances nach oben oder unten skalieren. Skalierungsvorgänge bei der Datenverarbeitung dauern in der Regel nur wenige Minuten.
- Amazon DocumentDB wird in Amazon Virtual Private Cloud (Amazon VPC) ausgeführt, sodass Sie Ihre Datenbank in Ihrem eigenen virtuellen Netzwerk isolieren können. Sie können auch Firewalleinstellungen so konfigurieren, dass der Netzwerkzugriff auf Ihren Cluster gesteuert wird.
- Amazon DocumentDB überwacht kontinuierlich den Zustand Ihres Clusters. Bei einem Instance-Ausfall startet Amazon DocumentDB die Instance und die zugehörigen Prozesse automatisch neu. Amazon DocumentDB erfordert keine Wiederholung der Datenbank-Redo-Logs zur Wiederherstellung nach einem Absturz, wodurch die Neustartzeiten erheblich reduziert werden. Amazon DocumentDB isoliert auch den Datenbank-Cache vom Datenbankprozess, sodass der Cache einen Instance-Neustart überstehen kann.
- Bei einem Instance-Ausfall automatisiert Amazon DocumentDB das Failover auf eines von bis zu 15 Amazon DocumentDB-Replikaten, die Sie in anderen Availability Zones erstellen. Wenn keine Replikate bereitgestellt wurden und ein Fehler auftritt, versucht Amazon DocumentDB automatisch, eine neue Amazon DocumentDB-Instance zu erstellen.
- Die Backup-Funktion in Amazon DocumentDB ermöglicht die point-in-time Wiederherstellung für Ihren Cluster. Diese Funktion ermöglicht Ihnen, Ihren Cluster zu jeder Sekunde innerhalb der Aufbewahrungsfrist bis zu den letzten 5 Minuten wiederherzustellen. Sie können den Aufbewahrungszeitraum für automatische Backups auf maximal 35 Tage festlegen. Automatisierte Backups werden in Amazon Simple Storage Service (Amazon S3) gespeichert, das für eine

Haltbarkeit von 99,999999999 % konzipiert ist. Amazon DocumentDB-Backups sind automatisch, inkrementell und kontinuierlich und haben keine Auswirkungen auf die Cluster-Leistung.

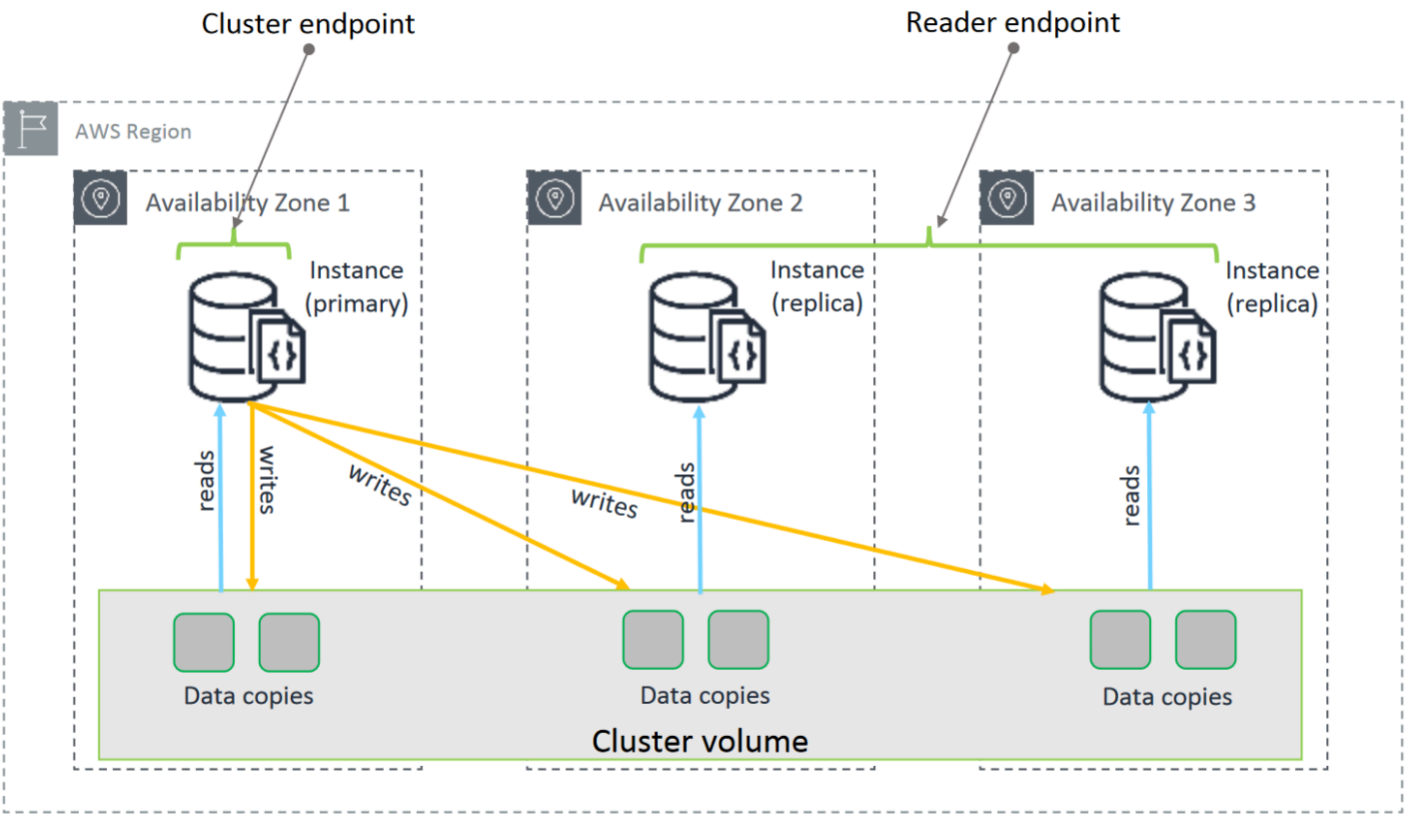
- Mit Amazon DocumentDB können Sie Ihre Datenbanken mit Schlüsseln verschlüsseln, die Sie über AWS Key Management Service () erstellen und steuernAWS KMS. Auf einem Datenbank-Cluster, der mit Amazon DocumentDB-Verschlüsselung ausgeführt wird, werden Daten verschlüsselt, die im Ruhezustand im zugrunde liegenden Speicher gespeichert sind. Die automatischen Sicherungen, Snapshots und Replicas im gleichen Cluster werden ebenfalls verschlüsselt.

Wenn Sie noch keine Erfahrung mit -AWSServices haben, verwenden Sie die folgenden Ressourcen, um mehr zu erfahren:

- AWS bietet Services für Datenverarbeitung, Datenbanken, Speicher, Analysen und andere Funktionen. Eine Übersicht über alle AWS Services finden Sie unter Cloud Computing mit Amazon Web Services.
- AWS bietet eine Reihe von Datenbankdiensten an. Hinweise dazu, welcher Service für Ihre Umgebung am besten geeignet ist, finden Sie unter Datenbanken auf AWS.

Cluster

Ein Cluster besteht aus 0 bis 16 Instances und einem Cluster-Speicher-Volume, das die Daten für diese Instances verwaltet. Alle Schreibvorgänge erfolgen über die primäre Instance. Alle Instances (primäre und Replicas) unterstützen Lesevorgänge. Die Daten des Clusters werden im Cluster-Volume gespeichert, mit Kopien in drei verschiedenen Availability Zones.



Instance-basierte Amazon DocumentDB-5.0-Cluster unterstützen zwei Speicherkonfigurationen für einen Datenbank-Cluster: Amazon DocumentDB-Standard und Amazon DocumentDB-E/A-optimiert. Weitere Informationen finden Sie unter Speicherkonfigurationen für Amazon DocumentDB-Cluster.

Instances

Eine Amazon DocumentDB-Instance ist eine isolierte Datenbankumgebung in der Cloud. Eine Instance kann mehrere von Benutzern erstellte Datenbanken enthalten. Sie können eine Instance erstellen und ändern, indem Sie die AWS Management Consoleoder AWS CLI verwenden.

Die Rechen- und Speicherkapazität einer Instance wird durch ihre Instance-Klasse bestimmt. Sie können die Instance auswählen, die Ihren Anforderungen am besten entspricht. Wenn sich Ihre Anforderungen im Laufe der Zeit ändern, können Sie eine andere Instance-Klasse wählen. Spezifikationen für DB-Instance-Klassen finden Sie unter Instance-Klassen-Spezifikationen

Amazon DocumentDB-Instances werden nur in der Amazon-VPC-Umgebung ausgeführt. Amazon VPC gibt Ihnen die Kontrolle über Ihre virtuelle Netzwerkumgebung: Sie können Ihren eigenen IP-Adressbereich auswählen, Subnetze erstellen und Routing- und Zugriffskontrolllisten (ACLs) konfigurieren.

Bevor Sie Amazon DocumentDB-Instances erstellen können, müssen Sie einen Cluster erstellen, der die Instances enthält.

Nicht alle Instance-Klassen werden in allen Regionen unterstützt. Die folgende Tabelle gibt an, welche Instance-Klassen von in den jeweiligen Regionen unterstützt werden.

Unterstützte Instance-Klassen nach Region

Region	R6G	R5	R4	T4G	T3
USA Ost (Ohio)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA Ost (Nord-Virginia)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA West (Oregon)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Südamerika (São Paulo)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hongkong)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hyderabad)		Unterstützt			Unterstützt
Asien-Pazifik (Mumbai)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Seoul)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Sydney)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Singapur)	Unterstützt	Unterstützt		Unterstützt	Unterstützt

Region	R6G	R5	R4	T4G	T3
Asien-Pazifik (Tokio)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Kanada (Zentral)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Frankfurt)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Irland)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Europa (London)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Milan)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Paris)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Region China (Peking)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
China (Ningxia)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (USA-West)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (USA-Ost)	Unterstützt	Unterstützt		Unterstützt	Unterstützt

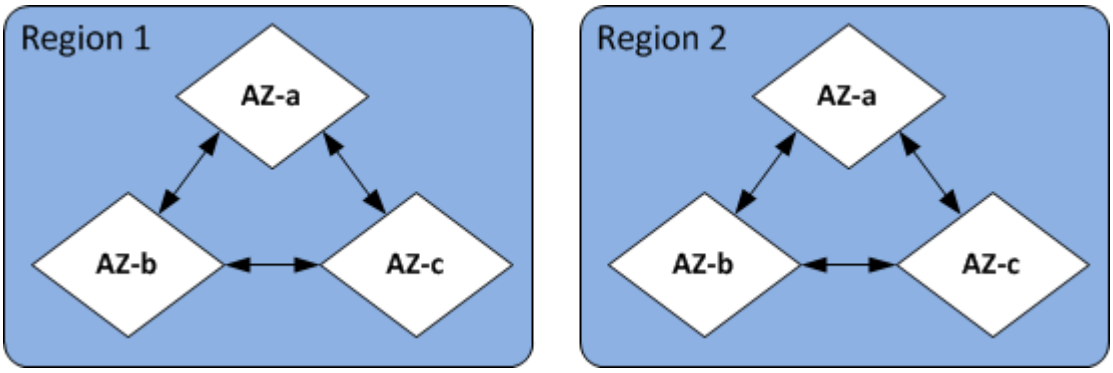
Regionen und Availability Zones

Regionen und Availability Zones definieren die physischen Standorte und Instances Ihres Clusters.

Regionen

AWS Cloud-Computing-Ressourcen befinden sich in hochverfügbaren Rechenzentrumszentren in verschiedenen Regionen der Welt (z. B. Nordamerika, Europa oder Asien). Jeder Rechenzentrumsstandort wird als Region bezeichnet.

Jede AWS-Region ist darauf ausgelegt, vollständig von den anderen AWS-Regionen getrennt zu sein. Innerhalb jeder gibt es mehrere Availability Zones (Verfügbarkeitszonen). Durch das Starten Ihrer Knoten in verschiedenen Availability Zones können Sie eine größtmögliche Fehlertoleranz zu erreichen. Das folgende Diagramm zeigt einen allgemeinen Überblick über die Funktionsweise von AWS Regionen und Availability Zones.



Availability Zones

Jede AWS Region enthält mehrere unterschiedliche Standorte, die als Availability Zones bezeichnet werden. Jede Availability Zone wurde so konzipiert, dass sie von Fehlern in anderen Availability Zones isoliert ist und eine kostengünstige Netzwerkverbindung mit geringer Latenz zu anderen Availability Zones in derselben Region bereitstellt. Indem Instances für einen bestimmten Cluster in mehreren Availability Zones gestartet werden, können Sie Ihre Anwendungen vor dem unwahrscheinlichen Fall des Fehlschlagens einer Availability Zone schützen.

Die Amazon DocumentDB-Architektur trennt Speicher und Datenverarbeitung. Für die Speicherebene repliziert Amazon DocumentDB sechs Kopien Ihrer Daten über drei AWS Availability Zones hinweg. Wenn Sie beispielsweise einen Amazon DocumentDB-Cluster in einer Region starten, die nur zwei Availability Zones unterstützt, wird Ihr Datenspeicher auf sechs Arten über drei Availability Zones repliziert, Ihre Datenverarbeitungs-Instances sind jedoch nur in zwei Availability Zones verfügbar.

In der folgenden Tabelle ist die Anzahl der Availability Zones aufgeführt, die Sie in einer bestimmten verwenden können, AWS-Region um Rechen-Instances für Ihren Cluster bereitzustellen.

Name der Region	Region	Availability Zones (Datenverarbeitung)
USA Ost (Ohio)	`us-east-2`	3
USA Ost (Nord-Virginia)	`us-east-1`	6
USA West (Oregon)	`us-west-2`	4
Südamerika (São Paulo)	`sa-east-1`	3
Asien-Pazifik (Hongkong)	`ap-east-1`	3
Asien-Pazifik (Hyderabad)	`ap-south-2`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	4
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Kanada (Zentral)	`ca-central-1`	3
Region China (Peking)	`cn-north-1`	3
China (Ningxia)	`cn-northwest-1`	3

Name der Region	Region	Availability Zones (Datenverarbeitung)
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
AWS GovCloud (USA-West)	`us-gov-west-1`	3
AWS GovCloud (USA-Ost)	`us-gov-east-1`	3

Amazon DocumentDB – Preise

Amazon DocumentDB-Cluster werden auf der Grundlage der folgenden Komponenten abgerechnet:

- Instance-Stunden (pro Stunde) – Basierend auf der Instance-Klasse der Instance (z. B. `db.r5.xlarge`). Die Preise werden auf Stundenbasis aufgeführt, aber Rechnungen werden jetzt auf die Sekunde genau kalkuliert und zeigen die Zeiten im Dezimalformat an. Amazon DocumentDB-Nutzung wird in Schritten von einer Sekunde abgerechnet, wobei mindestens 10 Minuten betragen. Weitere Informationen finden Sie unter Verwalten von Instance-Klassen.
- E/A-Anforderungen (pro 1 Million Anforderungen pro Monat) – Gesamtzahl der Speicher-E/A-Anforderungen, die Sie in einem Abrechnungszeitraum stellen.
- Backup-Speicher (pro GiB pro Monat) – Backup-Speicher ist der Speicher, der automatisierten Datenbanksicherungen und allen aktiven Datenbank-Snapshots zugeordnet ist, die Sie erstellt haben. Wenn Sie die Aufbewahrungszeit Ihrer Backups erhöhen oder zusätzliche Datenbank-Snapshots erstellen, belegt Ihre Datenbank dementsprechend mehr Backup-Speicher. Der Backup-Speicher wird in GB-Monaten abgerechnet, die sekundengenaue Abrechnung wird hier nicht angewandt. Weitere Informationen finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.

- Datenübertragung (pro GB) – Datenübertragung in und aus Ihrer Instance vom oder in das Internet oder andere -AWSRegionen.

Ausführliche Informationen finden Sie unter Amazon DocumentDB – Preise.

Kostenlose Testversion

Sie können Amazon DocumentDB kostenlos mit der 1-monatigen kostenlosen Testversion testen. Weitere Informationen finden Sie unter Kostenlose Testversion in Amazon DocumentDB – Preise oder in der kostenlosen Testversion von Amazon DocumentDB – Häufig gestellte Fragen.

Überwachen

Es gibt verschiedene Möglichkeiten, die Leistung und den Zustand einer Instance zu überwachen. Sie können den kostenlosen Amazon- CloudWatch Service verwenden, um die Leistung und den Zustand einer Instance zu überwachen. Sie finden Leistungsdiagramme in der Amazon DocumentDB-Konsole. Sie können Amazon DocumentDB-Ereignisse abonnieren, um benachrichtigt zu werden, wenn Änderungen an einer Instance, einem Snapshot, einer Parametergruppe oder einer Sicherheitsgruppe auftreten.

Weitere Informationen finden Sie hier:

- Überwachen von Amazon DocumentDB mit CloudWatch
- ProtokolDB-API-API-API-API-API-APIAWS CloudTrail

Schnittstellen

Es gibt mehrere Möglichkeiten, mit Amazon DocumentDB zu interagieren, einschließlich AWS Management Console und AWS CLI.

AWS Management Console

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

AWS CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Ihre Amazon DocumentDB-Cluster und -Instances zu verwalten. Mit minimaler Konfiguration können Sie die gesamte Funktionalität der Amazon DocumentDB-Konsole von Ihrem bevorzugten Terminalprogramm aus nutzen.

- Informationen zum Installieren der AWS CLI finden Sie unter Installieren der AWS-Befehlszeilenschnittstelle.
- Informationen zur Verwendung von AWS CLI für Amazon DocumentDB finden Sie in der -AWSBefehlszeilenschnittstellenreferenz für Amazon DocumentDB.

Die mongo-Shell

Um eine Verbindung zu Ihrem Cluster herzustellen, um Dokumente in Ihren Datenbanken zu erstellen, zu lesen, zu aktualisieren und zu löschen, können Sie die `mongo` Shell mit Amazon DocumentDB verwenden. Informationen zum Herunterladen und Installieren der `mongo` 4.0-Shell finden Sie unter Schritt 4: Installieren der mongo-Shell.

MongoDB-Treiber

Zum Entwickeln und Schreiben von Anwendungen in einem Amazon DocumentDB-Cluster können Sie auch die MongoDB-Treiber mit Amazon DocumentDB verwenden.

Die nächsten Themen

In den vorherigen Abschnitten wurden Sie mit den grundlegenden Infrastrukturkomponenten vertraut gemacht, die Amazon DocumentDB bietet. Was sollten Sie als nächstes tun? Lesen Sie je nach Ihren Umständen eines der folgenden Themen, um loszulegen:

- Beginnen Sie mit Amazon DocumentDB, indem Sie einen Cluster und eine Instance mit erstellenAWS CloudFormationAmazon DocumentDB-Schnellstart mit AWS CloudFormation.
- Beginnen Sie mit Amazon DocumentDB, indem Sie einen Cluster und eine Instance mit den Anweisungen in unserem erstellenHandbuch „Erste Schritte“.
- Beginnen Sie mit Amazon DocumentDB, indem Sie mithilfe der Anweisungen unter einen elastischen Cluster erstellenErste Schritte mit elastischen Amazon DocumentDB-Clustern.

- Migrieren Sie Ihre MongoDB-Implementierung zu Amazon DocumentDB anhand der Anleitung unter Migration zu Amazon DocumentDB

Amazon DocumentDB: Funktionsweise

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein vollständig verwalteter, MongoDB-kompatibler Datenbankservice. Mit Amazon DocumentDB können Sie denselben Anwendungscode ausführen und dieselben Treiber und Tools verwenden, die Sie mit MongoDB verwenden. Amazon DocumentDB ist mit MongoDB 3.6, 4.0 und 5.0 kompatibel.

Themen

Wenn Sie Amazon DocumentDB verwenden, erstellen Sie zunächst einen Cluster . Ein DB-Cluster besteht aus null oder mehreren Datenbank-Instances und einem Cluster-Volume, das die Daten für diese Instances verwaltet. Ein Amazon DocumentDB-Cluster-Volume ist ein virtuelles Datenbankspeicher-Volume, das sich über mehrere Availability Zones erstreckt. Jede Availability Zone verfügt über eine Kopie der Cluster-Daten.

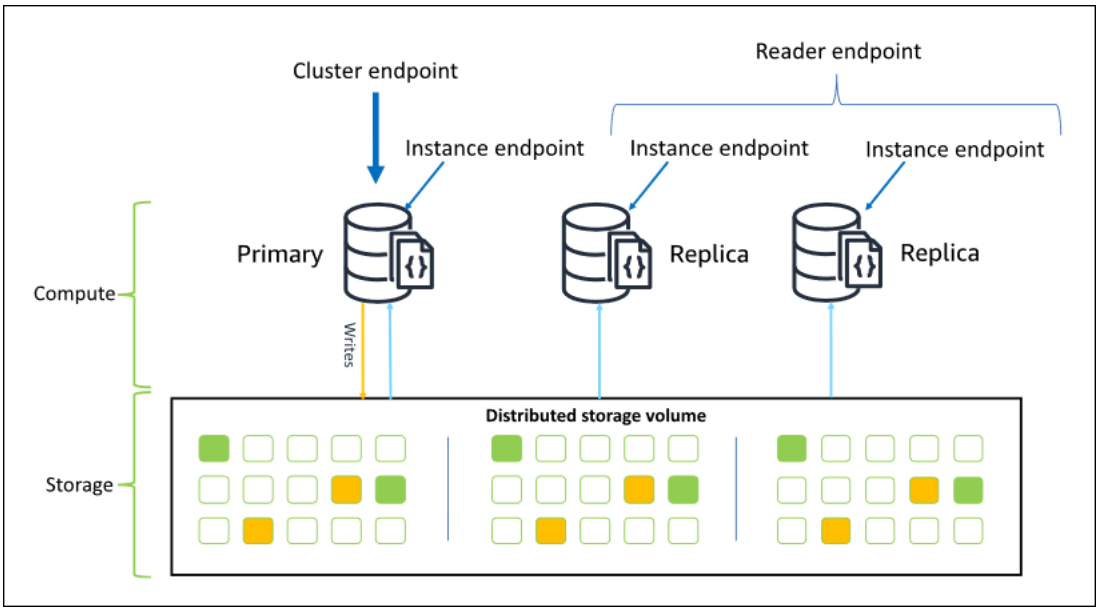
Ein Amazon DocumentDB-Cluster besteht aus zwei Komponenten:

- Cluster-Volume – Verwendet einen cloudnativen Speicherservice, um Daten auf sechs Arten über drei Availability Zones hinweg zu replizieren, was einen äußerst dauerhaften und verfügbaren Speicher bietet. Ein Amazon DocumentDB-Cluster hat genau ein Cluster-Volume, das bis zu 128 TiB Daten speichern kann.
- Instances – Stellen Sie die Rechenleistung für die Datenbank, das Schreiben von Daten in und das Lesen von Daten aus dem Cluster-Speicher-Volume bereit. Ein Amazon DocumentDB-Cluster kann 0–16 Instances haben.

Instances erfüllen eine von zwei Rollen:

- Primäre Instance – Unterstützt Lese- und Schreibvorgänge und führt alle Datenänderungen am Cluster-Volume durch. Jeder Amazon DocumentDB-Cluster verfügt über eine primäre Instance.
- Replikat-Instance – Unterstützt nur Lesevorgänge. Ein Amazon DocumentDB-Cluster kann zusätzlich zur primären Instance bis zu 15 Replikate haben. Die Verwendung mehrerer Replikate ermöglicht es Ihnen, die Leseauslastungen zu verteilen. Darüber hinaus erhöhen Sie durch die Platzierung von Replikaten in separaten Availability Zones auch die Cluster-Verfügbarkeit.

Das folgende Diagramm veranschaulicht die Beziehung zwischen dem Cluster-Volume, der primären Instance und Replikaten in einem Amazon DocumentDB-Cluster:



Cluster-Instances müssen nicht von derselben Instance-Klasse sein. Sie können beliebig bereitgestellt und beendet werden. Mit dieser Architektur können Sie die Rechenkapazität Ihres Clusters unabhängig von seiner Storage-Funktionalität skalieren.

Wenn Ihre Anwendung Daten in die Primär-Instance schreibt, schreibt diese die Daten dauerhaft in das Cluster-Volume. Anschließend repliziert es den Status dieses Schreibvorgangs (nicht die Daten) in jedes aktive Replikat. Amazon DocumentDB-Replikate nehmen nicht an der Verarbeitung von Schreibvorgängen teil, daher sind Amazon DocumentDB-Replikate für die Leseskalierung von Vorteil. Lesevorgänge aus Amazon DocumentDB-Replikaten sind letztendlich mit einer minimalen Replikatverzögerung konsistent – in der Regel weniger als 100 Millisekunden, nachdem die primäre Instance die Daten geschrieben hat. Lesezugriffe von den Replikaten werden garantiert in der Reihenfolge gelesen, in der sie auf die primäre Instance geschrieben wurden.

Die Replikationsverzögerung hängt von der Rate der Datenänderung ab. Perioden mit hoher Schreibaktivität können die Replikationsverzögerung erhöhen. Weitere Informationen finden Sie in den `ReplicationLag`-Metriken unter Amazon DocumentDB-Metriken.

Amazon DocumentDB-Endpunkte

Amazon DocumentDB bietet mehrere Verbindungsoptionen für eine Vielzahl von Anwendungsfällen. Um eine Verbindung zu einer Instance in einem Amazon DocumentDB-Cluster herzustellen, geben Sie den Endpunkt der Instance an. Ein Endpunkt ist eine Host-Adresse und eine Portnummer, getrennt durch einen Doppelpunkt.

Es wird empfohlen, dass Sie mithilfe des Clusterendpunkts und im Replikatsatzmodus eine Verbindung mit dem Cluster herstellen (siehe Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz), es sei denn, es liegt ein bestimmter Anwendungsfall für die Verbindung mit dem Reader-Endpunkt oder einem Instanceendpunkt vor. Um Anforderungen an Ihre Replikate weiterzuleiten, wählen Sie eine Treibereinstellung für die Leseeinstellung aus, die die Leseskalierung maximiert und gleichzeitig die Anforderungen für die Lesekonsistenz Ihrer Anwendung erfüllt. Die Leseeinstellung `secondaryPreferred` ermöglicht Replica-Lesevorgänge, sodass die primäre Instance produktiver sein kann.

Die folgenden Endpunkte sind in einem Amazon DocumentDB-Cluster verfügbar.

Cluster-Endpunkt

Der Cluster-Endpunkt verbindet sich mit der aktuellen primären Instance Ihres Clusters. Der Cluster-Endpunkt kann für Lese- und Schreibvorgänge verwendet werden. Ein Amazon DocumentDB-Cluster hat genau einen Cluster-Endpunkt.

Der Cluster-Endpunkt bietet Failover-Support für Lese-/Schreibverbindungen zum Cluster. Wenn die aktuelle primäre Instance Ihres Clusters ausfällt und Ihr Cluster mindestens eine aktive Read Replica hat, leitet der Cluster-Endpunkt Verbindungsanforderungen automatisch an eine neue primäre Instance weiter. Wenn Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen, empfehlen wir Ihnen, über den Cluster-Endpunkt und im Replikat-Set-Modus eine Verbindung zu Ihrem Cluster herzustellen (siehe Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz).

Im Folgenden finden Sie ein Beispiel für einen Amazon DocumentDB-Cluster-Endpunkt:

```
sample-cluster.cluster-123456789012.us-east-1.docdb.amazonaws.com:27017
```

Im Folgenden finden Sie ein Beispiel für eine Verbindungszeichenfolge für diesen Cluster-Endpunkt:

```
mongodb://username:password@sample-cluster.cluster-123456789012.us-
east-1.docdb.amazonaws.com:27017
```

Informationen zum Suchen der Endpunkte eines Clusters finden Sie unter Suchen der Endpunkte eines Clusters.

Reader-Endpunkt

Der Reader-Endpunkt agiert als Load-Balancer für schreibgeschützte Verbindungen für alle verfügbaren Replikate in Ihrem Cluster. Der Versuch, einen Schreibvorgang über eine Verbindung zum Reader-Endpunkt durchzuführen, führt zu einem Fehler. Ein Amazon DocumentDB-Cluster hat genau einen Reader-Endpunkt.

Wenn der Cluster nur eine (primäre) Instance enthält, verbindet sich der Reader-Endpunkt mit der primären Instance. Wenn Sie Ihrem Amazon DocumentDB-Cluster eine Replikat-Instance hinzufügen, öffnet der Reader-Endpunkt schreibgeschützte Verbindungen zum neuen Replikat, nachdem es aktiv ist.

Im Folgenden finden Sie ein Beispiel für einen Reader-Endpunkt für einen Amazon DocumentDB-Cluster:

```
sample-cluster.cluster-ro-123456789012.us-east-1.docdb.amazonaws.com:27017
```

Im Folgenden finden Sie ein Beispiel für eine Verbindungszeichenfolge unter Verwendung eines Reader-Endpunkts:

```
mongodb://username:password@sample-cluster.cluster-ro-123456789012.us-
east-1.docdb.amazonaws.com:27017
```

Der Reader-Endpunkt verteilt nur die Last der Read-only-Verbindungen – nicht die der Leseanforderungen. Wenn einige Reader-Endpunktverbindungen stärker genutzt werden als andere, sind Ihre Leseanforderungen möglicherweise nicht gleichmäßig zwischen Instances im Cluster verteilt. Es wird empfohlen, dass Sie zum Verteilen von Anforderungen eine Verbindung zum Clusterendpunkt als Replikatsatz herstellen und die Lesevorstellungsoption secondaryPreferred nutzen.

Informationen zum Suchen der Endpunkte eines Clusters finden Sie unter Suchen der Endpunkte eines Clusters.

Instance-Endpunkt

Ein Instance-Endpunkt verbindet sich mit einer bestimmten Instance innerhalb Ihres Clusters. Der Instance-Endpunkt für die aktuelle Primär-Instance kann für Lese- und Schreibvorgänge verwendet werden. Der Versuch, Schreiboperationen auf einen Instance-Endpunkt für ein Lesereplikat durchzuführen, führt jedoch zu einem Fehler. Ein Amazon DocumentDB-Cluster hat einen Instance-Endpunkt pro aktiver Instance.

Ein Instance-Endpunkt bietet für Szenarien, in denen der Cluster-Endpunkt oder der Lese-Endpunkt möglicherweise nicht geeignet ist, direkte Kontrolle über Verbindungen zu einer bestimmten Instance. Ein Beispiel für einen Anwendungsfall ist die Bereitstellung für einen periodischen Read-Only-Analyse-Workload. Sie können eine larger-than-normal Replikat-Instance bereitstellen, eine direkte Verbindung mit der neuen größeren Instance mit ihrem Instance-Endpunkt herstellen, die Analyseabfragen ausführen und dann die Instance beenden. Die Verwendung des Instance-Endpunkts verhindert, dass sich der Analyseverkehr auf andere Cluster-Instances auswirkt.

Im Folgenden finden Sie ein Beispiel für einen Instance-Endpunkt für eine einzelne Instance in einem Amazon DocumentDB-Cluster:

```
sample-instance.123456789012.us-east-1.docdb.amazonaws.com:27017
```

Im Folgenden finden Sie ein Beispiel für eine Verbindungszeichenfolge mit diesem Instance-Endpunkt:

```
mongodb://username:password@sample-instance.123456789012.us-
east-1.docdb.amazonaws.com:27017
```

> **Note**
>
> Die Rolle einer Instance als "Primär" oder "Replikat" kann sich aufgrund eines Failover-Ereignisses ändern. Ihre Anwendungen sollten niemals davon ausgehen, dass ein bestimmter Instance-Endpunkt die primäre Instance ist. Es wird nicht empfohlen, eine Verbindung zu Instance-Endpunkten für Produktionsanwendungen herzustellen. Stattdessen wird empfohlen, dass Sie mithilfe des Clusterendpunkts und im Replikatsatzmodus eine Verbindung zum Cluster herstellen (siehe Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz). Weitere Informationen zur erweiterten Kontrolle der Instance-Failover-Priorität finden Sie unter Grundlegendes zur Fehlertoleranz des Amazon DocumentDB-Clusters.

Informationen zum Suchen der Endpunkte eines Clusters finden Sie unter Suchen nach dem Endpunkt einer Instance.

Replikatsatzmodus

Sie können im Replikatsatzmodus eine Verbindung zu Ihrem Amazon DocumentDB-Cluster-Endpunkt herstellen, indem Sie den Namen des Replikatsatzes angeben`rs0`. Die Verbindung im Replikatsatzmodus bietet die Möglichkeit, die Optionen Read Concern, Write Concern und Read Preference festzulegen. Weitere Informationen finden Sie unter Lesekonsistenz.

Im Folgenden finden Sie ein Beispiel für eine Verbindungszeichenfolge, die im Replikatsatzmodus verbunden ist:

```
mongodb://username:password@sample-cluster.cluster-123456789012.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0
```

Wenn Sie eine Verbindung im Replikat-Set-Modus herstellen, wird Ihr Amazon DocumentDB-Cluster Ihren Treibern und Clients als Replikat-Set angezeigt. Instances, die Ihrem Amazon DocumentDB-Cluster hinzugefügt und daraus entfernt wurden, werden automatisch in der Replikatsatzkonfiguration wiedergegeben.

Jeder Amazon DocumentDB-Cluster besteht aus einem einzelnen Replikatsatz mit dem Standardnamen `rs0`. Der Name des Replikatsatzes kann nicht geändert werden.

Die Verbindung mit dem Cluster-Endpunkt im Replikatsatzmodus ist die empfohlene Methode für den allgemeinen Gebrauch.

> **Note**
>
> Alle Instances in einem Amazon DocumentDB-Cluster überwachen denselben TCP-Port für Verbindungen.

TLS Support

Weitere Informationen zum Herstellen einer Verbindung mit Amazon DocumentDB mithilfe von Transport Layer Security (TLS) finden Sie unter Datenverschlüsselung während der Übertragung.

Amazon DocumentDB-Speicher

Amazon DocumentDB-Daten werden in einem Cluster-Volume gespeichert. Dabei handelt es sich um ein einzelnes virtuelles Volume, das Solid-State-Laufwerke (SSDs) verwendet. Ein Cluster-Volume besteht aus sechs Kopien Ihrer Daten, die automatisch über mehrere Availability Zones in einem einzigen repliziert werdenAWS-Region. Diese Replikation trägt dazu bei, dass Ihre Daten sehr langlebig sind und weniger Datenverlust möglich ist. Sie trägt außerdem dazu bei, dass Ihr Cluster während eines Failovers besser verfügbar ist, da Kopien Ihrer Daten bereits in anderen Availability Zones vorhanden sind. Diese Kopien können weiterhin Datenanforderungen an die Instances in Ihrem Amazon DocumentDB-Cluster verarbeiten.

Informationen zur Abrechnung des -Datenspeichers

Amazon DocumentDB erhöht automatisch die Größe eines Cluster-Volumes, wenn die Datenmenge zunimmt. Ein Amazon DocumentDB-Cluster-Volume kann auf eine maximale Größe von 128 TiB anwachsen. Ihnen wird jedoch nur der Speicherplatz in Rechnung gestellt, den Sie in einem Amazon DocumentDB-Cluster-Volume verwenden. Beginnend mit Amazon DocumentDB 4.0 verringert sich der insgesamt zugewiesene Speicherplatz um einen vergleichbaren Betrag, wenn Daten entfernt werden, z. B. durch Löschen einer Sammlung oder eines Indexes. Auf diese Weise können Sie die Speicherkosten senken, indem Sie Sammlungen, Indizes und Datenbanken löschen, die Sie nicht mehr benötigen. Wenn mit Amazon DocumentDB 3.6 Daten entfernt werden, z. B. durch Löschen einer Sammlung oder eines Index, bleibt der insgesamt zugewiesene Speicherplatz gleich. Der freie Speicherplatz wird automatisch wiederverwendet, wenn das Datenvolumen in Zukunft zunimmt.

> **Note**
>
> Mit Amazon DocumentDB 3.6 basieren die Speicherkosten auf dem Speicher „High water mark“ (der maximalen Menge, die dem Amazon DocumentDB-Cluster zu einem beliebigen Zeitpunkt zugewiesen wurde). Sie können Kosten verwalten, indem Sie ETL-Methoden vermeiden, die große Mengen an temporären Informationen erzeugen oder große Mengen neuer Daten laden, bevor Sie nicht benötigte ältere Daten entfernen. Wenn das Entfernen von Daten aus einem Amazon DocumentDB-Cluster zu einer erheblichen Menge an zugewiesenem, aber ungenutztem Speicherplatz führt, erfordert das Zurücksetzen des hohen Wasserzeichens, dass ein logischer Datenabbild durchgeführt und mithilfe eines Tools wie `mongodump` oder auf einem neuen Cluster wiederhergestellt wird`mongorestore`. Das Erstellen und Wiederherstellen eines Snapshots führt nicht zur Reduzierung des

zugeteilten Speichers, da das physische Layout des zugrunde liegenden Speichers im wiederhergestellten Snapshot unverändert bleibt.

> **Note**
>
> Die Verwendung von Dienstprogrammen wie `mongodump` und `mongorestore` verursachen E/A-Gebühren basierend auf der Größe der Daten, die gelesen und auf das Speicher-Volume geschrieben werden.

Informationen zu Amazon DocumentDB-Datenspeicher und I/O-Preisen finden Sie unter Amazon DocumentDB-Preise (mit MongoDB-Kompatibilität) und Häufig FAQs Fragen zu Preisen.

Amazon DocumentDB-Replikation

In einem Amazon DocumentDB-Cluster stellt jede Replikat-Instance einen unabhängigen Endpunkt bereit. Diese Replikat-Endpunkte bieten Lesezugriff auf die Daten im Cluster-Volume. Mit ihnen können Sie die Leselast für Ihre Daten über mehrere replizierte Instances hinweg skalieren. Sie tragen auch dazu bei, die Leistung von Datenlesevorgängen zu verbessern und die Verfügbarkeit der Daten in Ihrem Amazon DocumentDB-Cluster zu erhöhen. Amazon DocumentDB-Replikate sind ebenfalls Failover-Ziele und werden schnell hochgestuft, wenn die primäre Instance für Ihren Amazon DocumentDB-Cluster ausfällt.

Zuverlässigkeit von Amazon DocumentDB

Amazon DocumentDB ist darauf ausgelegt, zuverlässig, dauerhaft und fehlertolerant zu sein. (Um die Verfügbarkeit zu verbessern, sollten Sie Ihren Amazon DocumentDB-Cluster so konfigurieren, dass er über mehrere Replikat-Instances in verschiedenen Availability Zones verfügt.) Amazon DocumentDB enthält mehrere automatische Funktionen, die es zu einer zuverlässigen Datenbanklösung machen.

Automatische Reparatur des Speicherplatzes

Amazon DocumentDB verwaltet mehrere Kopien Ihrer Daten in drei Availability Zones, wodurch die Wahrscheinlichkeit eines Datenverlusts aufgrund eines Speicherausfalls erheblich reduziert wird. Amazon DocumentDB erkennt automatisch Fehler im Cluster-Volume. Wenn ein Segment eines Cluster-Volumes ausfällt, repariert Amazon DocumentDB das Segment sofort. Es verwendet die Daten der anderen Volumes, aus denen sich das Cluster-Volumen zusammensetzt, um

sicherzustellen, dass die Daten im reparierten Segment aktuell sind. Dadurch vermeidet Amazon DocumentDB Datenverlust und reduziert die Notwendigkeit einer point-in-time Wiederherstellung, um eine Wiederherstellung nach einem Instance-Ausfall durchzuführen.

Überlebensfähiges Cache-Warming

Amazon DocumentDB verwaltet seinen Seiten-Cache in einem von der Datenbank getrennten Prozess, sodass der Seiten-Cache unabhängig von der Datenbank überleben kann. Im unwahrscheinlichen Fall eines Datenbankausfalls, bleibt der Seiten-Cache im Arbeitsspeicher. Auf diese Weise wird sichergestellt, dass der Pufferpool beim Neustart der Datenbank mit dem aktuellen Zustand vorbereitet wird.

Wiederherstellung nach einem Ausfall

Amazon DocumentDB ist so konzipiert, dass die Wiederherstellung nach einem Absturz nahezu sofort erfolgt und Ihre Anwendungsdaten weiterhin bereitgestellt werden. Amazon DocumentDB führt die Wiederherstellung nach einem Absturz asynchron auf parallelen Threads durch, sodass Ihre Datenbank nach einem Absturz fast sofort geöffnet und verfügbar ist.

Ressourcenverwaltung

Amazon DocumentDB schützt Ressourcen, die für die Ausführung kritischer Prozesse im Service benötigt werden, z. B. Zustandsprüfungen. Dazu drosselt Amazon DocumentDB Anfragen und wenn eine Instance einen hohen Speicherdruck hat. Infolgedessen können einige Operationen in die Warteschlange gestellt werden, um zu warten, bis der Speicherdruck abnimmt. Wenn der Speicherdruck weiterhin besteht, kann es bei Operationen in der Warteschlange zu einem Timeout kommen. Sie können mit den folgenden CloudWatch Metriken überwachen, ob die Service-Drosselungsvorgänge aufgrund von geringem Speicher gedrosselt werden oder nicht: `LowMemThrottleQueueDepth`, `LowMemThrottleMaxQueueDepth`, `LowMemNumOperationsThrottled`, `LowMemNumOperationsTimedOut`. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch. Wenn Sie aufgrund der LowMem CloudWatch Metriken anhaltenden Speicherdruck auf Ihrer Instance feststellen, empfehlen wir Ihnen, Ihre Instance hochzuskalieren, um zusätzlichen Speicher für Ihre Workload bereitzustellen.

Leseeinstellungsoptionen

Amazon DocumentDB verwendet einen cloudnativen Shared Storage-Service, der Daten sechsmal über drei Availability Zones repliziert, um ein hohes Maß an Haltbarkeit zu gewährleisten. Amazon DocumentDB verlässt sich nicht darauf, Daten auf mehrere Instances zu replizieren, um eine

Haltbarkeit zu erreichen. Die Daten Ihres Clusters sind beständig, unabhängig davon, ob sie eine einzelne Instance oder 15 Instances enthalten.

Beständigkeit von Schreibvorgängen

Amazon DocumentDB verwendet ein eindeutiges, verteiltes, fehlertolerantes, selbstverarbeitendes Speichersystem. Dieses System repliziert sechs Kopien (V=6) Ihrer Daten in drei AWS Availability Zones, um eine hohe Verfügbarkeit und Haltbarkeit zu gewährleisten. Beim Schreiben von Daten stellt Amazon DocumentDB sicher, dass alle Schreibvorgänge auf den meisten Knoten dauerhaft aufgezeichnet werden, bevor der Schreibvorgang an den Client bestätigt wird. Wenn Sie einen MongoDB-Replikatsatz mit drei Knoten ausführen, `{w:3, j:true}` würde die Verwendung eines Schreibproblems von beim Vergleich mit Amazon DocumentDB die bestmögliche Konfiguration ergeben.

Schreibvorgänge in einen Amazon DocumentDB-Cluster müssen von der Writer-Instance des Clusters verarbeitet werden. Der Versuch, in einen Reader zu schreiben, führt zu einem Fehler. Ein bestätigter Schreibvorgang von einer primären Amazon DocumentDB-Instance ist beständig und kann nicht rückgängig gemacht werden. Amazon DocumentDB ist standardmäßig sehr langlebig und unterstützt keine nicht dauerhafte Schreiboption. Sie können die Zuverlässigkeitsstufe (d. h. die Option Write Concern) nicht ändern. Amazon DocumentDB ignoriert w=alles und ist effektiv w: 3 und j: true. Sie können es nicht reduzieren.

Da Speicher und Datenverarbeitung in der Amazon DocumentDB-Architektur getrennt sind, ist ein Cluster mit einer einzigen Instance äußerst langlebig. Die Zuverlässigkeit wird auf der Speicherschicht geregelt. Daher erzielen ein Amazon DocumentDB-Cluster mit einer einzelnen Instance und einer mit drei Instances das gleiche Maß an Haltbarkeit. Sie können Ihren Cluster für Ihren speziellen Anwendungsfall konfigurieren und gleichzeitig für eine hohe Datenbeständigkeit sorgen.

Schreibvorgänge in einen Amazon DocumentDB-Cluster sind in einem einzigen Dokument atomar.

Amazon DocumentDB unterstützt die `wtimeout` Option nicht und gibt keinen Fehler zurück, wenn ein Wert angegeben wird. Schreibvorgänge auf die primäre Amazon DocumentDB-Instance werden garantiert nicht auf unbestimmte Zeit blockiert.

Isolierung von Lesevorgängen

Lesevorgänge aus einer Amazon DocumentDB-Instance geben nur Daten zurück, die dauerhaft sind, bevor die Abfrage beginnt. Lesezugriffe geben niemals Daten zurück, die nach Beginn der Ausführung der Abfrage geändert wurden. Auch "Dirty-Reads" sind unter keinen Umständen möglich.

Lesekonsistenz

Daten, die aus einem Amazon DocumentDB-Cluster gelesen werden, sind dauerhaft und werden nicht zurückgesetzt. Sie können die Lesekonsistenz für Amazon DocumentDB-Lesevorgänge ändern, indem Sie die Lesepräferenz für die Anforderung oder Verbindung angeben. Amazon DocumentDB unterstützt keine nicht dauerhafte Leseoption.

Lesevorgänge aus der primären Instance eines Amazon DocumentDB-Clusters sind unter normalen Betriebsbedingungen stark konsistent und haben read-after-write Konsistenz. Tritt zwischen dem Schreiben und dem nachfolgenden Lesen ein Failover-Ereignis auf, kann das System kurzzeitig einen nicht "Strongly Consistent"-Wert zurückgeben. Alle Lesezugriffe auf einer gelesenen Replik sind "Eventually Consistent" und geben die Daten in der gleichen Reihenfolge zurück, oft mit weniger als 100 ms Replikationsverzögerung.

Amazon DocumentDB-Lesepräferenzen

Amazon DocumentDB unterstützt das Festlegen einer Lesepräferenzoption nur beim Lesen von Daten vom Cluster-Endpunkt im Replikat-Set-Modus. Das Festlegen einer Lesepräferenzoption wirkt sich darauf aus, wie Ihr MongoDB-Client oder -Treiber Leseanforderungen an Instances in Ihrem Amazon DocumentDB-Cluster weiterleitet. Sie können Leseeinstellungen für eine bestimmte Abfrage oder als allgemeine Option in Ihrem MongoDB-Treiber festlegen. (Lesen Sie in der Dokumentation Ihres Clients oder Treibers nach, wie Sie eine Leseeinstellung festlegen können.)

Wenn Ihr Client oder Treiber im Replikatsatzmodus keine Verbindung zu einem Amazon DocumentDB-Cluster-Endpunkt herstellt, ist das Ergebnis der Angabe einer Lesepräferenz nicht definiert.

Amazon DocumentDB unterstützt nicht das Festlegen von Tag-Sets als Lesepräferenz.

Unterstützte Leseeinstellungsoptionen

- **`primary`**– Die Angabe einer `primary` Lesepräferenz trägt dazu bei, dass alle Lesevorgänge an die primäre Instance des Clusters weitergeleitet werden. Wenn die primäre Instance nicht verfügbar ist, schlägt der Lesevorgang fehl. Eine `primary` Lesepräferenz führt zu read-after-write Konsistenz und eignet sich für Anwendungsfälle, die read-after-write Konsistenz gegenüber hoher Verfügbarkeit und Leseskalierung priorisieren.

  Im folgenden Beispiel wird die Leseeinstellung `primary` angegeben:

  ```
  db.example.find().readPref('primary')
  ```

- **`primaryPreferred`**– Die Angabe einer `primaryPreferred` Lesepräferenz leitet Lesevorgänge im normalen Betrieb an die primäre Instance weiter. Wenn es ein primäres Failover gibt, leitet der Client Anfragen an ein Replikat weiter. Eine `primaryPreferred` Lesepräferenz führt während des normalen Betriebs zu read-after-write Konsistenz und schließlich zu konsistenten Lesevorgängen während eines Failover-Ereignisses. Eine `primaryPreferred` Lesepräferenz ist für Anwendungsfälle geeignet, die read-after-write Konsistenz gegenüber der Leseskalierung priorisieren, aber dennoch eine hohe Verfügbarkeit erfordern.

  Im folgenden Beispiel wird die Leseeinstellung `primaryPreferred` angegeben:

  ```
  db.example.find().readPref('primaryPreferred')
  ```

- **`secondary`**– Die Angabe einer `secondary` Lesepräferenz stellt sicher, dass Lesevorgänge nur an ein Replikat und niemals an die primäre Instance weitergeleitet werden. Wenn es in einem Cluster keine Replikat-Instances gibt, schlägt die Leseanforderung fehl. Eine `secondary` Lesepräferenz führt zu Eventually-Consistent-Lesevorgängen und eignet sich für Anwendungsfälle, die den Schreibdurchsatz der primären Instance gegenüber hoher Verfügbarkeit und read-after-write Konsistenz priorisieren.

  Im folgenden Beispiel wird die Leseeinstellung `secondary` angegeben:

  ```
  db.example.find().readPref('secondary')
  ```

- **`secondaryPreferred`**– Die Angabe einer `secondaryPreferred` Lesepräferenz stellt sicher, dass Lesevorgänge an ein Lesereplikat weitergeleitet werden, wenn ein oder mehrere Replikate aktiv sind. Wenn es in einem Cluster keine aktiven Replikat-Instances gibt, wird die Leseanforderung an die primäre Instance weitergeleitet. Die Leseeinstellung `secondaryPreferred` ergibt Eventually Consistent-Lesezugriffe, wenn der Lesezugriff durch ein Read Replica bedient wird. Dies führt zu read-after-write Konsistenz, wenn der Lesevorgang von der primären Instance bedient wird (wobei Failover-Ereignisse ausgeschlossen werden). Eine `secondaryPreferred` Lesepräferenz ist für Anwendungsfälle geeignet, die Leseskalierung und hohe Verfügbarkeit gegenüber read-after-write Konsistenz priorisieren.

Im folgenden Beispiel wird die Leseeinstellung `secondaryPreferred` angegeben:

```
db.example.find().readPref('secondaryPreferred')
```

- **`nearest`**– Die Angabe einer `nearest` Lesepräferenz leitet Lesevorgänge ausschließlich basierend auf der gemessenen Latenz zwischen dem Client und allen Instances im Amazon DocumentDB-Cluster weiter. Die Leseeinstellung `nearest` ergibt Eventually Consistent-Lesezugriffe, wenn der Lesezugriff durch ein Read Replica bedient wird. Dies führt zu read-after-write Konsistenz, wenn der Lesevorgang von der primären Instance bedient wird (wobei Failover-Ereignisse ausgeschlossen werden). Eine `nearest` Lesepräferenz ist für Anwendungsfälle geeignet, in denen priorisiert wird, die niedrigstmögliche Leselatenz und hohe Verfügbarkeit gegenüber read-after-write Konsistenz und Leseskalierung zu erreichen.

  Im folgenden Beispiel wird die Leseeinstellung `nearest` angegeben:

  ```
  db.example.find().readPref('nearest')
  ```

Hochverfügbarkeit

Amazon DocumentDB unterstützt hochverfügbare Clusterkonfigurationen, indem Replikate als Failover-Ziele für die primäre Instance verwendet werden. Wenn die primäre Instance ausfällt, wird ein Amazon DocumentDB-Replikat als neue primäre Instance hochgestuft, mit einer kurzen Unterbrechung, während der Lese- und Schreibanforderungen an die primäre Instance mit einer Ausnahme fehlschlagen.

Wenn Ihr Amazon DocumentDB-Cluster keine Replikate enthält, wird die primäre Instance während eines Fehlers neu erstellt. Das Hochstufen eines Amazon DocumentDB-Replikats ist jedoch viel schneller als das Neuerstellen der primären Instance. Daher empfehlen wir, ein oder mehrere Amazon DocumentDB-Replikate als Failover-Ziele zu erstellen.

Replikate, die als Failover-Ziele verwendet werden sollen, sollten dieselbe DB-Instance-Klasse haben wie die primäre Instance. Sie sollten von der primären Instance in verschiedenen Availability Zones bereitgestellt werden. Sie können steuern, welche Replikate als Failover-Ziele bevorzugt werden. Bewährte Methoden zur Konfiguration von Amazon DocumentDB für hohe Verfügbarkeit finden Sie unter Grundlegendes zur Fehlertoleranz des Amazon DocumentDB-Clusters.

Skalierung von Lesevorgängen

Amazon DocumentDB-Replikate eignen sich ideal für die Leseskalierung. Sie sind in Ihrem Cluster-Volume vollständig auf Lesevorgänge ausgerichtet, d. h. Replikate verarbeiten keine Schreibvorgänge. Die Datenreplikation geschieht innerhalb des Cluster-Volumes und nicht zwischen Instances. Deshalb sind die Replikat-Ressourcen auf die Verarbeitung von Abfragen ausgelegt und nicht auf das Schreiben und Replizieren von Daten.

Wenn Ihre Anwendung mehr Lesekapazität benötigt, können Sie Ihrem Cluster schnell (in der Regel in weniger als zehn Minuten) eine Replik hinzufügen. Wenn Ihr Lesekapazitätsbedarf sinkt, können Sie nicht benötigte Replikate entfernen. Mit Amazon DocumentDB-Replikaten zahlen Sie nur für die benötigte Lesekapazität.

Amazon DocumentDB unterstützt die clientseitige Leseskalierung durch die Verwendung von Lesepräferenzoptionen. Weitere Informationen finden Sie unter Amazon DocumentDB-Lesepräferenzen.

TTL löscht

Löschungen aus einem TTL-Indexbereich über einen Hintergrundprozess erfolgen nach dem Best-Effort-Prinzip und können nicht für einen bestimmten Zeitrahmen garantiert werden. Faktoren wie Instance-Größe, Instance-Ressourcenauslastung, Dokumentgröße und Gesamtdurchsatz können sich auf den Zeitpunkt einer TTL-Löschung auswirken.

Wenn der TTL-Monitor Ihre Dokumente löscht, entstehen bei jeder Löschung E/A-Kosten, was den Rechnungsbetrag erhöht. Wenn die Durchsatz- und TTL-Löschraten steigen, sollten Sie einen höheren Rechnungsbetrag erwarten, da die E/A-Nutzung steigt.

Wenn Sie einen TTL-Index für eine vorhandene Sammlung erstellen, müssen Sie alle abgelaufenen Dokumente löschen, bevor Sie den Index erstellen. Die aktuelle TTL-Implementierung ist für das Löschen eines kleinen Bruchteils von Dokumenten in der Sammlung optimiert, was in der Regel der Fall ist, wenn TTL bei der Sammlung von Anfang an aktiviert wurde, und kann zu höheren IOPS als nötig führen, wenn eine große Anzahl von Dokumenten auf einmal gelöscht werden muss.

Wenn Sie keinen TTL-Index zum Löschen von Dokumenten erstellen möchten, können Sie stattdessen Dokumente nach Zeit in Sammlungen segmentieren und diese Sammlungen einfach löschen, wenn die Dokumente nicht mehr benötigt werden. Sie können beispielsweise eine Sammlung pro Woche erstellen und sie löschen, ohne dass Ihnen I/O-Kosten entstehen. Dies kann deutlich kostengünstiger sein als die Verwendung eines TTL-Index.

Fakturierbare Ressourcen

Identifizieren fakturierbarer Amazon DocumentDB-Ressourcen

Als vollständig verwalteter Datenbankservice berechnet Amazon DocumentDB Gebühren für Instances, Speicher, E/A-Operationen, Backups und Datenübertragungen. Weitere Informationen finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) – Preise.

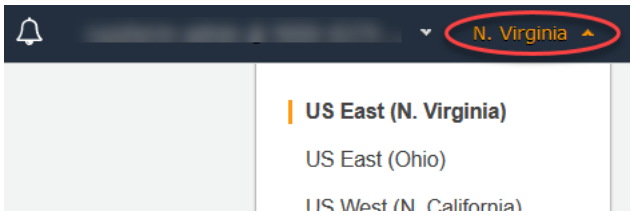
Um kostenpflichtige Ressourcen in Ihrem Konto zu identifizieren und möglicherweise zu löschen, können Sie die AWS Management Console oder AWS CLI verwenden.

Verwenden des AWS Management Console

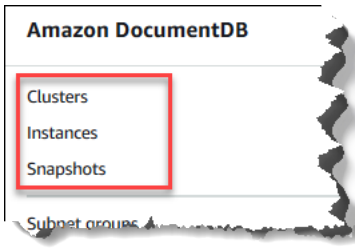
Mit der können AWS Management ConsoleSie die Amazon DocumentDB-Cluster, -Instances und -Snapshots ermitteln, die Sie für eine bestimmte bereitgestellt habenAWS-Region.

So ermitteln Sie Cluster, Instances und Snapshots:

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Um abrechenbare Ressourcen in einer anderen Region als Ihrer Standardregion zu ermitteln, wählen Sie in der oberen rechten Ecke des Bildschirms die aus, AWS-Region nach der Sie suchen möchten.



3. Wählen Sie im Navigationsbereich die Art der kostenpflichtigen Ressource aus: Clusters (Cluster), Instances oder Snapshots.



4. Im rechten Bereich werden alle bereitgestellten Cluster, Instances oder Snapshots für die Region aufgelistet. Für Cluster, Instances und Snapshots werden Gebühren berechnet.

Verwenden des AWS CLI

Mit der können AWS CLISie die Amazon DocumentDB-Cluster, -Instances und -Snapshots ermitteln, die Sie für eine bestimmte bereitgestellt habenAWS-Region.

So ermitteln Sie Cluster und Instances:

Der folgende Code listet Ihre gesamten Cluster und Instances für die angegebene Region auf. Wenn Sie nach Clustern und Instances in Ihrer Standardregion suchen möchten, können Sie den Parameter `--region` weglassen.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --region us-east-1 \
    --query 'DBClusters[?Engine==`docdb`]' | \
       grep -e "DBClusterIdentifier" -e "DBInstanceIdentifier"
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --region us-east-1 ^
    --query 'DBClusters[?Engine==`docdb`]' | ^
       grep -e "DBClusterIdentifier" -e "DBInstanceIdentifier"
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
"DBClusterIdentifier": "docdb-2019-01-09-23-55-38",
        "DBInstanceIdentifier": "docdb-2019-01-09-23-55-38",
        "DBInstanceIdentifier": "docdb-2019-01-09-23-55-382",
"DBClusterIdentifier": "sample-cluster",
"DBClusterIdentifier": "sample-cluster2",
```

So ermitteln Sie Snapshots:

Der folgende Code listet Ihre gesamten Snapshots für die angegebene Region auf. Wenn Sie in Ihrer Standardregion nach Snapshots suchen möchten, können Sie den Parameter `--region` weglassen.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-snapshots \
  --region us-east-1 \
  --query 'DBClusterSnapshots[?Engine==`docdb`].
[DBClusterSnapshotIdentifier,SnapshotType]'
```

Für Windows:

```
aws docdb describe-db-cluster-snapshots ^
  --region us-east-1 ^
  --query 'DBClusterSnapshots[?Engine==`docdb`].
[DBClusterSnapshotIdentifier,SnapshotType]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "rds:docdb-2019-01-09-23-55-38-2019-02-13-00-06",
        "automated"
    ],
    [
        "test-snap",
        "manual"
    ]
]
```

Sie müssen nur `manual`-Snapshots löschen. `Automated`-Snapshots werden gelöscht, wenn Sie den Cluster löschen.

Löschen unerwünschter kostenpflichtiger Ressourcen

Um einen Cluster zu löschen, müssen Sie zunächst alle Instances im Cluster löschen.

- Weitere Informationen zum Löschen von Instances finden Sie unter Löschen einer Amazon DocumentDB-Instance .

> **Important**
>
> Auch wenn Sie die Instances in einem Cluster löschen, wird Ihnen die mit diesem Cluster verbundene Speicher- und Sicherungsnutzung in Rechnung gestellt. Um alle Kosten zu stoppen, müssen Sie auch Ihren Cluster und manuelle Snapshots löschen.

- Weitere Informationen zum Löschen von Clustern finden Sie unter Löschen eines Amazon DocumentDB-Clusters.
- Weitere Informationen zum Löschen manueller Snapshots finden Sie unter Löschen eines Cluster-Snapshots.

Was ist eine Dokumentdatenbank?

Einige Entwickler betrachten ihr Datenmodell nicht als normalisierte Zeilen und Spalten. In der Regel werden Daten auf Anwendungsebene als JSON-Dokument dargestellt, da es für Entwickler intuitiver ist, ihr Datenmodell als Dokument zu betrachten.

Die Popularität von Dokumentdatenbanken ist gestiegen, weil Sie mit ihnen Daten in einer Datenbank erhalten können, indem Sie das gleiche Dokumentmodellformat verwenden, das Sie in Ihrem Anwendungscode verwenden. Dokumentdatenbanken bieten leistungsfähige und intuitive APIs für eine flexible und agile Entwicklung.

Themen

- Anwendungsfälle der Dokumentdatenbank
- Informationen zu Dokumenten
- Arbeiten mit Dokumenten

Anwendungsfälle der Dokumentdatenbank

Ihr Anwendungsfall bestimmt, ob Sie eine Dokumentendatenbank oder eine andere Art von Datenbank für die Verwaltung Ihrer Daten benötigen. Dokumentdatenbanken sind nützlich für Workloads, die ein flexibles Schema für eine schnelle, iterative Entwicklung benötigen. Im Folgenden finden Sie einige Beispiele für Anwendungsfälle, bei denen Dokumentendatenbanken erhebliche Vorteile bieten können:

Themen

- Benutzerprofile
- Big-Data in Echtzeit
- Content-Management

Benutzerprofile

Da Dokumentdatenbanken über ein flexibles Schema verfügen, können sie Dokumente speichern, die unterschiedliche Attribute und Datenwerte aufweisen. Dokumentdatenbanken sind eine praktische Lösung für Online-Profile, in denen verschiedene Benutzer unterschiedliche Arten von Informationen bereitstellen. Mit Hilfe einer Dokumentdatenbank können Sie das Profil jedes Benutzers effizient speichern, indem Sie nur die Attribute speichern, die für jeden Benutzer spezifisch sind.

Angenommen, ein Benutzer entscheidet sich dafür, Informationen in seinem Profil hinzuzufügen oder zu entfernen. In diesem Fall könnte ihr Dokument leicht durch eine aktualisierte Version ersetzt werden, die alle kürzlich hinzugefügten Attribute und Daten enthält oder alle neu weggelassenen Attribute und Daten auslässt. Dokumentdatenbanken bewältigen diese Individualität und Fluidität auf einfache Weise.

Big-Data in Echtzeit

In der Vergangenheit wurde die Fähigkeit, Informationen aus Betriebsdaten zu extrahieren, durch die Tatsache behindert, dass betriebliche Datenbanken und Analysedatenbanken in verschiedenen Umgebungen — betrieblicher bzw. geschäftlicher Berichterstattung — gepflegt wurden. Die Fähigkeit, operative Informationen in Echtzeit zu extrahieren, ist in einem hart umkämpften Geschäftsumfeld von entscheidender Bedeutung. Durch die Verwendung von Dokumentendatenbanken kann ein Unternehmen Betriebsdaten aus jeder beliebigen Quelle speichern und verwalten und gleichzeitig die Daten zur Analyse an die BI-Engine seiner Wahl weiterleiten. Es ist nicht erforderlich, dass zwei Umgebungen vorhanden sein müssen.

Content-Management

Um Inhalte effektiv zu verwalten, müssen Sie in der Lage sein, Inhalte aus einer Vielzahl von Quellen zu sammeln, zu aggregieren und dann an den Client zu liefern. Aufgrund ihres flexiblen Schemas sind Dokumentdatenbanken ideal für die Erfassung und Speicherung jeglicher Art von Daten. Mit ihnen können Sie neue Arten von Inhalten erstellen und integrieren, einschließlich benutzergenerierter Inhalte wie Bilder, Kommentare und Videos.

Informationen zu Dokumenten

Dokumentdatenbanken werden zum Speichern von semistrukturierten Daten als Dokument verwendet, anstatt Daten über mehrere Tabellen mit jeweils einer eindeutigen und festen Struktur wie in einer relationalen Datenbank zu normalisieren. Dokumente, die in einer Dokumentdatenbank

gespeichert sind, verwenden verschachtelte Schlüssel-Werte-Paare, um die Struktur oder das Schema des Dokuments bereitzustellen. Es können jedoch verschiedene Arten von Dokumenten in derselben Dokumentdatenbank gespeichert werden, wodurch die Anforderung erfüllt wird, ähnliche Daten in unterschiedlichen Formaten zu verarbeiten. Da beispielsweise jedes Dokument selbstbeschreibend ist, können die in diesem Abschnitt beschriebenen JSON-kodierten Dokumente für einen Online-Shop, die unter dem Thema Beispieldokumente in einer Dokumentdatenbank beschrieben werden, in derselben Dokumentendatenbank gespeichert werden.

Themen

SQL gegenüber Nicht-relationale Terminologie

Die folgende Tabelle vergleicht die von Dokumentendatenbanken (MongoDB) verwendete Terminologie mit der von SQL-Datenbanken verwendeten Terminologie.

SQL	MongoDB
Tabelle	Sammlung
Zeile	Dokument
Spalte	Feld
Primärschlüssel	ObjectId
Index	Index
Anzeigen	Anzeigen
Verschachtelte(s) Tabelle oder Objekt	Eingebettetes Dokument
Array	Array

Einfache Dokumente

Alle Dokumente in einer Dokumentendatenbank sind selbstbeschreibend. In dieser Dokumentation werden JSON-ähnlich formatierte Dokumente verwendet, Sie können aber auch andere Verschlüsselungsmethoden einsetzen.

Ein einfaches Dokument hat ein oder mehrere Felder, die alle auf der gleichen Ebene innerhalb des Dokuments liegen. Im folgenden Beispiel sind die Felder `SSN`, `LName`, `FName`, `DOB`, `Street`, `City`, `State-Province`, `PostalCode` und `Country` alle Elemente auf derselben Ebene im Dokument.

```
{
   "SSN": "123-45-6789",
   "LName": "Rivera",
   "FName": "Martha",
   "DOB": "1992-11-16",
   "Street": "125 Main St.",
   "City": "Anytown",
   "State-Province": "WA",
   "PostalCode": "98117",
   "Country": "USA"
}
```

Wenn Informationen in einem einfachen Dokument organisiert sind, wird jedes Feld einzeln verwaltet. Um die Adresse einer Person abzurufen, müssen Sie `Street`, `City`, `State-Province`, `PostalCode` und `Country` als einzelne Datenelemente abrufen.

Eingebettete Dokumente

Ein komplexes Dokument organisiert seine Daten, indem es eingebettete Dokumente innerhalb des Dokuments erstellt. Eingebettete Dokumente helfen bei der Verwaltung von Daten in Gruppierungen und als einzelne Datenelemente, je nachdem, was im jeweiligen Fall effizienter ist. Mit dem vorhergehenden Beispiel könnten Sie ein `Address`-Dokument in das Hauptdokument einbetten. Dadurch ergibt sich die folgende Dokumentstruktur:

```
{
   "SSN": "123-45-6789",
   "LName": "Rivera",
   "FName": "Martha",
   "DOB": "1992-11-16",
   "Address":
```

```
   {
       "Street": "125 Main St.",
       "City": "Anytown",
       "State-Province": "WA",
       "PostalCode": "98117",
       "Country": "USA"
   }
}
```

Sie können jetzt auf die Daten im Dokument als einzelne Felder zugreifen (`"SSN":`), als eingebettetes Dokument (`"Address":`) oder als Mitglied eines eingebetteten Dokuments (`"Address":{"Street":}`) enthalten.

Beispieldokumente in einer Dokumentdatenbank

Da jedes Dokument in einer Dokumentdatenbank selbstbeschreibend ist, kann die Struktur von Dokumenten innerhalb einer Dokumentdatenbank unterschiedlich sein. Die folgenden beiden Dokumente, eines für ein Buch und eines für eine Zeitschrift, sind strukturell unterschiedlich. Beide können sich jedoch in der gleichen Dokumentdatenbank befinden.

Im Folgenden finden Sie ein Beispiel-Buch-Dokument:

```
{
    "_id" : "9876543210123",
    "Type": "book",
    "ISBN": "987-6-543-21012-3",
    "Author":
    {
        "LName":"Roe",
        "MI": "T",
        "FName": "Richard"
    },
    "Title": "Understanding Document Databases"
}
```

Im Folgenden finden Sie ein Beispiel für ein Zeitschriften-Dokument mit zwei Artikeln:

```
{
    "_id" : "0123456789012",
    "Publication": "Programming Today",
    "Issue":
```

```
    {
        "Volume": "14",
        "Number": "09"
    },
    "Articles" : [
        {
            "Title": "Is a Document Database Your Best Solution?",
            "Author":
            {
                "LName": "Major",
                "FName": "Mary"
            }
        },
        {
            "Title": "Databases for Online Solutions",
            "Author":
            {
                "LName": "Stiles",
                "FName": "John"
            }
        }
    ],
    "Type": "periodical"
}
```

Vergleichen Sie die Struktur dieser beiden Dokumente. Bei einer relationalen Datenbank benötigen Sie entweder getrennte Tabellen "Zeitschriften" und "Bücher" oder eine einzelne Tabelle mit unbenutzten Feldern wie "Publikation", "Ausgabe", "Artikel" und "MI" als `null` Werte. Da Dokumentendatenbanken halbstrukturiert sind und jedes Dokument seine eigene Struktur definiert, können diese beiden Dokumente ohne `null`-Felder gleichzeitig in derselben Dokumentendatenbank vorhanden sein. Dokumentdatenbanken sind gut geeignet, um mit lückenhaften Daten umzugehen.

Die Entwicklung auf Basis einer Dokumentdatenbank ermöglicht eine schnelle, iterative Entwicklung. Denn Sie können die Datenstruktur eines Dokuments dynamisch ändern, ohne das Schema für die gesamte Sammlung ändern zu müssen. Dokumentdatenbanken eignen sich hervorragend für agile Entwicklungen und sich dynamisch verändernde Umgebungen.

Grundlegendes zur Normalisierung in einer Dokumentdatenbank

Dokumentdatenbanken werden nicht normalisiert; in einem Dokument gefundene Daten können in einem anderen Dokument wiederholt werden. Darüber hinaus können einige Datenunterschiede zwischen den Dokumenten bestehen. Betrachten Sie beispielsweise das Szenario, in dem Sie einen

Einkauf in einem Online-Shop tätigen und alle Details Ihrer Einkäufe in einem einzigen Dokument gespeichert sind. Das Dokument könnte etwa so aussehen wie das folgende JSON-Dokument:

```
{
    "DateTime": "2018-08-15T12:13:10Z",
    "LName" : "Santos",
    "FName" : "Paul",
    "Cart" : [
        {
            "ItemId" : "9876543210123",
            "Description" : "Understanding Document Databases",
            "Price" : "29.95"
        },
        {
            "ItemId" : "0123456789012",
            "Description" : "Programming Today",
            "Issue": {
                "Volume": "14",
                "Number": "09"
            },
            "Price" : "8.95"
        },
        {
            "ItemId": "234567890-K",
            "Description": "Gel Pen (black)",
            "Price": "2.49"
        }
    ],
    "PaymentMethod" :
    {
        "Issuer" : "MasterCard",
        "Number" : "1234-5678-9012-3456"
    },
    "ShopperId" : "1234567890"
}
```

Alle diese Informationen werden als Dokument in einer Transaktionssammlung gespeichert. Später merken Sie, dass Sie vergessen haben, einen Artikel zu kaufen. Daher melden Sie sich erneut im gleichen Shop an und tätigen einen weiteren Kauf, der auch als ein weiteres Dokument in der Transaktionssammlung gespeichert wird.

```
{
```

```
    "DateTime": "2018-08-15T14:49:00Z",
    "LName" : "Santos",
    "FName" : "Paul",
    "Cart" : [
        {
            "ItemId" : "2109876543210",
            "Description" : "Document Databases for Fun and Profit",
            "Price" : "45.95"
        }
    ],
    "PaymentMethod" :
    {
        "Issuer" : "Visa",
        "Number" : "0987-6543-2109-8765"
    },
    "ShopperId" : "1234567890"
}
```

Beachten Sie die Redundanz zwischen diesen beiden Dokumenten - Ihrem Namen und Ihrer Käufer-ID (und, wenn Sie dieselbe Kreditkarte verwendet haben, Ihre Kreditkarteninformationen). Aber das ist in Ordnung, denn die Speicherung ist kostengünstig, und jedes Dokument zeichnet eine einzelne Transaktion vollständig auf, die mit einer einfachen Schlüssel-/Wertabfrage, die keine Verknüpfungen erfordert, schnell abgerufen werden kann.

Es besteht auch eine offensichtliche Diskrepanz zwischen den beiden Dokumenten - Ihren Kreditkarteninformationen. Dies ist nur eine scheinbare Diskrepanz, da es wahrscheinlich ist, dass Sie für jeden Kauf eine andere Kreditkarte verwendet haben. Jedes Dokument entspricht genau der Transaktion, die es dokumentiert.

Arbeiten mit Dokumenten

Als Dokumentdatenbank macht es Amazon DocumentDB einfach, JSON-Daten zu speichern, abzufragen und zu indizieren. In Amazon DocumentDB ist eine Sammlung analog zu einer Tabelle in einer relationalen Datenbank, nur dass es kein einzelnes Schema gibt, das auf alle Dokumente angewendet wird. Mit Sammlungen können Sie ähnliche Dokumente gruppieren und gleichzeitig alle in derselben Datenbank halten, ohne dass sie in ihrer Struktur identisch sein müssen.

Unter Verwendung der Beispieldokumente aus früheren Abschnitten ist es wahrscheinlich, dass Sie Sammlungen für `reading_material` und `office_supplies` haben werden. Es liegt in der Verantwortung Ihrer Software, die Zugehörigkeit eines Dokuments zu einer bestimmten Sammlung durchzusetzen.

Die folgenden Beispiele zeigen anhand der MongoDB-API, wie Sie Dokumente hinzufügen, abfragen, aktualisieren und löschen können.

Themen

Hinzufügen von Dokumenten

In Amazon DocumentDB wird eine Datenbank erstellt, wenn Sie einer Sammlung zum ersten Mal ein Dokument hinzufügen. In diesem Beispiel erstellen Sie eine Sammlung mit dem Namen `example` in der Datenbank `test`, die die Standarddatenbank ist, wenn Sie eine Verbindung mit einem Cluster herstellen. Da die Sammlung implizit beim Einfügen des ersten Dokuments erstellt wird, erfolgt keine Fehlerprüfung des Sammlungsnamens. Das heißt, ein Tippfehler im Sammlungsnamen, wie z. B. `eexample` statt `example` führt dazu, dass das Dokument erstellt und der Sammlung `eexample` und nicht der beabsichtigten Sammlung hinzugefügt wird. Die Fehlerüberprüfung muss von Ihrer Anwendung durchgeführt werden.

Die folgenden Beispiele verwenden die MongoDB-API zum Hinzufügen der Dokumente.

Themen

Hinzufügen eines einzelnen Dokuments

Um ein einzelnes Dokument zu einer Sammlung hinzuzufügen, verwenden Sie die Operation `insertOne( {} )` mit dem Dokument, das Sie der Sammlung hinzufügen möchten.

```
db.example.insertOne(
    {
        "Item": "Ruler",
        "Colors": ["Red","Green","Blue","Clear","Yellow"],
        "Inventory": {
            "OnHand": 47,
            "MinOnHand": 40
```

```
        },
        "UnitPrice": 0.89
    }
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "acknowledged" : true,
    "insertedId" : ObjectId("5bedafbcf65ff161707de24f")
}
```

Hinzufügen mehrerer Dokumente

Um mehrere Dokumente zu einer Sammlung hinzuzufügen, verwenden Sie die Operation `insertMany( [{},...,{}] )` mit einer Liste der Dokumente, die Sie der Sammlung hinzufügen möchten. Obwohl die Dokumente in dieser Liste unterschiedliche Schemata haben, können sie alle zur gleichen Sammlung hinzugefügt werden.

```
db.example.insertMany(
    [
        {
            "Item": "Pen",
            "Colors": ["Red","Green","Blue","Black"],
            "Inventory": {
                "OnHand": 244,
                "MinOnHand": 72
            }
        },
        {
            "Item": "Poster Paint",
            "Colors": ["Red","Green","Blue","Black","White"],
            "Inventory": {
                "OnHand": 47,
                "MinOnHand": 50
            }
        },
        {
            "Item": "Spray Paint",
            "Colors": ["Black","Red","Green","Blue"],
            "Inventory": {
                "OnHand": 47,
```

```
                "MinOnHand": 50,
                "OrderQnty": 36
            }
        }
    ]
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "acknowledged" : true,
    "insertedIds" : [
            ObjectId("5bedb07941ca8d9198f5934c"),
            ObjectId("5bedb07941ca8d9198f5934d"),
            ObjectId("5bedb07941ca8d9198f5934e")
    ]
}
```

Abfragen von Dokumenten

Manchmal müssen Sie möglicherweise den Bestand Ihres Online-Shops nachschlagen, damit Kunden das Angebot sehen und kaufen können. Die Abfrage einer Sammlung ist relativ einfach, unabhängig davon, ob Sie alle Dokumente in der Sammlung haben möchten oder nur die Dokumente, die ein bestimmtes Kriterium erfüllen.

Verwenden Sie die Operation `find()`, um Dokumente abzufragen. Der Befehl `find()` hat einen einzigen Dokumentenparameter, der die Kriterien für die Auswahl der zurückzugebenden Dokumente definiert. Die Ausgabe von `find()` ist ein Dokument, das als einzelne Textzeile ohne Zeilenumbrüche formatiert ist. Um das Ausgabedokument für eine bessere Lesbarkeit zu formatieren, verwenden Sie `find().pretty()`. Alle Beispiele in diesem Thema verwenden `.pretty()` zum Formatieren der Ausgabe.

Verwenden Sie die vier Dokumente, die Sie in die`example`Sammlung in den beiden vorangegangenen Übungen —`insertOne()`und`insertMany()`aus.

Themen

- Alle Dokumente einer Sammlung abrufen
- Dokumente abrufen, die einem Feldwert entsprechen
- Abrufen von Dokumenten, die einem eingebetteten Dokument entsprechen
- Abrufen von Dokumenten, die einem Feldwert in einem eingebetteten Dokument entsprechen

- Abrufen von Dokumenten, die einem Array entsprechen
- Abrufen von Dokumenten, die einem Wert in einem Array entsprechen
- Abrufen von Dokumenten mithilfe von Operatoren

Alle Dokumente einer Sammlung abrufen

Um alle Dokumente in Ihrer Sammlung abzurufen, verwenden Sie die Operation `find()` mit einem leeren Abfragedokument.

Die folgende Abfrage gibt alle Dokumente der Sammlung `example` zurück.

```
db.example.find( {} ).pretty()
```

Dokumente abrufen, die einem Feldwert entsprechen

Um alle Dokumente abzurufen, die mit einem Feld und einem Wert übereinstimmen, verwenden Sie die Operation `find()` mit einem Abfragedokument, das die entsprechenden Felder und Werte identifiziert.

Bei Verwendung der vorangegangenen Dokumente gibt diese Abfrage alle Dokumente zurück, bei denen das Feld "Item" (Element) "Pen" (Stift) entspricht.

```
db.example.find( { "Item": "Pen" } ).pretty()
```

Abrufen von Dokumenten, die einem eingebetteten Dokument entsprechen

Um alle Dokumente zu suchen, die mit einem eingebetteten Dokument übereinstimmen, verwenden Sie die Operation `find()` mit einem Abfragedokument, in dem der Name des eingebetteten Dokuments sowie alle Felder und Werte für dieses eingebettete Dokument angegeben werden.

Beim Vergleichen mit einem eingebetteten Dokument muss das eingebettete Dokument denselben Namen haben wie in der Abfrage. Zudem müssen die Felder und Werte im eingebetteten Dokument mit der Abfrage übereinstimmen.

Die folgende Abfrage gibt nur das Dokument "Poster Paint" zurück. Dies liegt daran, dass "Pen" über verschiedene Werte für "`OnHand`" und "`MinOnHand`" verfügt und "Spray Paint" ein weiteres Feld (`OrderQnty`) als das Abfragedokument besitzt.

```
db.example.find({"Inventory": {
```

```
    "OnHand": 47,
    "MinOnHand": 50 } } ).pretty()
```

Abrufen von Dokumenten, die einem Feldwert in einem eingebetteten Dokument entsprechen

Um alle Dokumente zu suchen, die mit einem eingebetteten Dokument übereinstimmen, verwenden Sie die Operation `find()` mit einem Abfragedokument, in dem der Name des eingebetteten Dokuments sowie alle Felder und Werte für dieses eingebettete Dokument angegeben werden.

Aufgrund der vorangegangenen Dokumente verwendet die folgende Abfrage "Punktnotation", um das eingebettete Dokument und die Felder von Interesse anzugeben. Jedes Dokument, das damit übereinstimmt, wird zurückgegeben, unabhängig davon, welche anderen Felder im eingebetteten Dokument vorhanden sind. Die Abfrage gibt "Poster Paint" und "Spray Paint" zurück, weil sie beide den angegebenen Feldern und Werten entsprechen.

```
db.example.find({"Inventory.OnHand": 47, "Inventory.MinOnHand": 50 }).pretty()
```

Abrufen von Dokumenten, die einem Array entsprechen

Um alle Dokumente zu finden, die einem Array entsprechen, verwenden Sie die Operation `find()` mit dem Namen des Arrays, an dem Sie interessiert sind, und allen Werten in diesem Array. Die Abfrage gibt alle Dokumente zurück, in denen sich ein Array mit diesem Namen befindet und in denen die Array-Werte identisch sind und die gleiche Reihenfolge wie in der Abfrage aufweisen.

Die folgende Abfrage gibt nur "Pen" zurück, da "Poster Paint" über eine zusätzlichen Farbe (White) verfügt und die Farben in "Spray Paint" in einer anderen Reihenfolge vorliegen.

```
db.example.find( { "Colors": ["Red","Green","Blue","Black"] } ).pretty()
```

Abrufen von Dokumenten, die einem Wert in einem Array entsprechen

Um alle Dokumente mit einem bestimmten Array-Wert zu finden, verwenden Sie die Operation `find()` mit dem Namen und Wert des Arrays, an dem Sie interessiert sind.

```
db.example.find( { "Colors": "Red" } ).pretty()
```

Bei der vorherigen Operation werden alle drei Dokumente zurückgegeben, da jedes davon ein Array mit dem Namen `Colors` und den Wert "`Red`" irgendwo im Array besitzt. Wenn Sie den Wert "`White`" angeben, gibt die Abfrage nur "Poster Paint" zurück.

Abrufen von Dokumenten mithilfe von Operatoren

Die folgende Abfrage gibt alle Dokumente zurück, in denen der Wert "`Inventory.OnHand`" kleiner als 50 ist.

```
db.example.find(
        { "Inventory.OnHand": { $lt: 50 } } )
```

Eine Liste der unterstützten Abfrageoperatoren finden Sie unter Abfrage- und Projektions-Operatoren.

Aktualisierung von Dokumenten

In der Regel sind Ihre Dokumente nicht statisch und werden als Teil Ihrer Anwendungs-Workflows aktualisiert. Die folgenden Beispiele zeigen einige der Möglichkeiten, wie Sie Dokumente aktualisieren können.

Um ein bestehendes Dokument zu aktualisieren, verwenden Sie die Operation`update()`. Die Operation `update()` besitzt zwei Dokumentenparameter. Das erste Dokument gibt an, welche Dokumente aktualisiert werden sollen. Das zweite Dokument gibt an, welche Aktualisierungen durchzuführen sind.

Wenn Sie ein vorhandenes Feld aktualisieren (unabhängig davon, ob das Feld ein einfaches Feld, ein Array oder ein eingebettetes Dokument ist) - geben Sie den Feldnamen und seine Werte an. Am Ende der Operation wirkt es so, als ob das Feld im alten Dokument durch das neue Feld und die Werte ersetzt wurde.

Themen

- Aktualisieren der Werte eines vorhandenen Feldes
- Hinzufügen eines neuen Feldes
- Ersetzen eines eingebetteten Dokuments
- Einfügen eines neuen Feldes in ein eingebettetes Dokument
- Entfernen eines Feldes aus einem Dokument
- Entfernen eines Felds aus mehreren Dokumenten

Aktualisieren der Werte eines vorhandenen Feldes

Verwenden Sie die folgenden vier Dokumente, die Sie zuvor hinzugefügt haben, für die folgenden Aktualisierungsoperationen.

```
{
    "Item": "Ruler",
    "Colors": ["Red","Green","Blue","Clear","Yellow"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 40
    },
    "UnitPrice": 0.89
},
{
    "Item": "Pen",
    "Colors": ["Red","Green","Blue","Black"],
    "Inventory": {
        "OnHand": 244,
        "MinOnHand": 72
    }
},
{
    "Item": "Poster Paint",
    "Colors": ["Red","Green","Blue","Black","White"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50
    }
},
{
    "Item": "Spray Paint",
    "Colors": ["Black","Red","Green","Blue"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50,
        "OrderQnty": 36
    }
}
```

So aktualisieren Sie ein einfaches Feld

Um ein einfaches Feld zu aktualisieren, verwenden Sie `update()` mit `$set`, um den Feldnamen und den neuen Wert anzugeben. Im folgenden Beispiel wird das `Item` von "Pen" in "Gel Pen" geändert.

```
db.example.update(
    { "Item" : "Pen" },
    { $set: { "Item": "Gel Pen" } }
```

```
)
```

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

```
{
    "Item": "Gel Pen",
    "Colors": ["Red","Green","Blue","Black"],
    "Inventory": {
        "OnHand": 244,
        "MinOnHand": 72
    }
}
```

So aktualisieren Sie ein Array

Das folgende Beispiel ersetzt das vorhandene Array von Farben mit einem neuen Array, das in der Liste der Farben `Orange` enthält und `White` weglässt. Die neue Liste von Farben liegt in der Reihenfolge vor, die in der Operation `update()` festgelegt wurde.

```
db.example.update(
    { "Item" : "Poster Paint" },
    { $set: { "Colors": ["Red","Green","Blue","Orange","Black"] } }
)
```

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

```
{
    "Item": "Poster Paint",
    "Colors": ["Red","Green","Blue","Orange","Black"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50
    }
}
```

Hinzufügen eines neuen Feldes

Um ein Dokument durch Hinzufügen eines oder mehrerer neuer Felder zu ändern, verwenden Sie die Operation `update()` mit einem Abfragedokument, das das einzufügende Dokument und die neuen Felder und Werte, die mit dem Operator `$set` eingefügt werden sollen, identifiziert.

Im folgenden Beispiel wird das Feld `UnitPrice` mit dem Wert `3.99` dem Dokument "Spray Paints" hinzugefügt. Beachten Sie, dass der Wert `3.99` numerisch ist und keine Zeichenfolge.

```
db.example.update(
    { "Item": "Spray Paint" },
    { $set: { "UnitPrice": 3.99 } }
)
```

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus (JSON-Format).

```
{
    "Item": "Spray Paint",
    "Colors": ["Black","Red","Green","Blue"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 50,
        "OrderQnty": 36
    },
    "UnitPrice": 3.99
}
```

Ersetzen eines eingebetteten Dokuments

Um ein Dokument durch Ersetzen eines eingebetteten Dokuments zu ändern, verwenden Sie die Operation `update()` mit Dokumenten, die das eingebettete Dokument und seine neuen Felder und Werte mit dem Operator `$set` identifizieren.

Anhand des folgenden Dokuments

```
db.example.insert({
    "DocName": "Document 1",
    "Date": {
        "Year": 1987,
        "Month": 4,
        "Day": 18
    }
})
```

So ersetzen Sie ein eingebettetes Dokument

Im folgenden Beispiel wird das aktuellen Datumsdokument durch ein neues ersetzt, das nur die Felder `Month` und `Day` besitzt; `Year` wurde entfernt.

```
db.example.update(
    { "DocName" : "Document 1" },
    { $set: { "Date": { "Month": 4, "Day": 18 } } }
)
```

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18
    }
}
```

Einfügen eines neuen Feldes in ein eingebettetes Dokument

So fügen Sie einem eingebetteten Dokument neue Felder hinzu

Um ein Dokument durch Hinzufügen eines oder mehrerer neuer Felder zu einem eingebetteten Dokument zu ändern, verwenden Sie die Operation `update()` mit Dokumenten, die das eingebettete Dokument identifizieren, und "Punktnotation" zum Angeben des eingebetteten Dokuments und der neuen Felder und Werte verwenden, die mit dem Operator `$set` eingefügt werden sollen.

Beim folgenden Dokument verwendet der folgende Code "Punktnotierung", um die Felder `DoW` und `Year` im eingebetteten `Date`-Dokument und `Words` im übergeordneten Dokument einzufügen.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18
    }
}
```

```
db.example.update(
    { "DocName" : "Document 1" },
    { $set: { "Date.Year": 1987,
              "Date.DoW": "Saturday",
              "Words": 2482 } }
```

```
)
```

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18,
        "Year": 1987,
        "DoW": "Saturday"
    },
    "Words": 2482
}
```

Entfernen eines Feldes aus einem Dokument

Um ein Dokument zu ändern, indem Sie ein Feld aus dem Dokument entfernen, verwenden Sie die Operation `update()` mit einem Abfragedokument, das das Dokument identifiziert, aus dem das Feld entfernt werden soll, und den Operator `$unset`, um das zu entfernende Feld anzugeben.

Das folgende Beispiel entfernt das Feld `Words` aus dem vorangegangen Dokument.

```
db.example.update(
    { "DocName" : "Document 1" },
    { $unset: { Words:1 } }
)
```

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

```
{
    "DocName": "Document 1",
    "Date": {
        "Month": 4,
        "Day": 18,
        "Year": 1987,
        "DoW": "Saturday"
    }
}
```

Entfernen eines Felds aus mehreren Dokumenten

Um ein Dokument zu ändern, indem ein Feld aus mehreren Dokumenten entfernt wird, verwenden Sie die Operation `update()` mit dem Operator `$unset` und der Option `multi`, die auf `true` festgelegt ist.

Im folgenden Beispiel wird das Feld `Inventory` aus allen Dokumenten in der Beispielsammlung entfernt. Wenn ein Dokument das Feld `Inventory` nicht hat, wird keine Aktion für dieses Dokument durchgeführt. Wenn `multi: true` weggelassen wird, wird die Aktion nur für das erste Dokument ausgeführt, das das Kriterium erfüllt.

```
db.example.update(
    {},
    { $unset: { Inventory:1 } },
    { multi: true }
)
```

Dokumente löschen

Um ein Dokument aus Ihrer Datenbank zu entfernen, verwenden Sie die Operation `remove()` und geben Sie an, welches Dokument Sie entfernen möchten. Der folgende Code entfernt "Gel Pen" aus Ihrer `example`-Sammlung.

```
db.example.remove( { "Item": "Gel Pen" } )
```

Um alle Dokumente aus Ihrer Datenbank zu entfernen, verwenden Sie die Operation `remove()` mit einer leeren Abfrage (s. unten).

```
db.example.remove( { } )
```

Erste Schritte mit Amazon DocumentDB

Es gibt viele Möglichkeiten, eine Verbindung herzustellen und mit Amazon DocumentDB zu beginnen. Wir haben diesen Leitfaden erstellt, da wir diese Möglichkeit als schnellste, einfachste und einfachste Möglichkeit für Benutzer gefunden haben, unsere leistungsstarke Dokumentdatenbank zu nutzen. In diesem Handbuch wird verwendetAWS Cloud9, ein webbasiertes Terminal, um Ihren Amazon DocumentDB-Cluster mithilfe der mongo-Shell direkt über die zu verbinden und abzufragenAWS Management Console. Neue Kunden, die für das AWS kostenlose Kontingent für berechtigt sind, können Amazon DocumentDB und AWS Cloud9 kostenlos nutzen. Wenn Ihre AWS Cloud9 Umgebung oder Ihr Amazon DocumentDB-Cluster -Ressourcen über das kostenlose Kontingent hinaus nutzt, werden Ihnen die normalen AWS -Gebühren für diese Ressourcen in Rechnung gestellt. In diesem Handbuch werden Sie in weniger als 15 Minuten mit Amazon DocumentDB beginnen.

> **Note**
>
> Die Anweisungen in diesem Handbuch beziehen sich speziell auf das Erstellen und Herstellen einer Verbindung mit Amazon DocumentDB-Clustern. Wenn Sie elastische Amazon DocumentDB-Cluster erstellen und eine Verbindung zu ihnen herstellen möchten, finden Sie weitere Informationen unter Erste Schritte mit elastischen Amazon DocumentDB-Clustern.

Themen

Wenn Sie lieber von Ihrem lokalen Computer aus eine Verbindung zu Ihrem Amazon DocumentDB herstellen möchten, indem Sie eine SSH-Verbindung zu einer Amazon EC2-Instance herstellen, lesen Sie bitte die Anweisungen Mit EC2 verbinden

Voraussetzungen

Bevor Sie Ihren ersten Amazon DocumentDB-Cluster erstellen, müssen Sie Folgendes tun:

Erstellen eines Amazon Web Services (AWS)-Kontos

Bevor Sie Amazon DocumentDB verwenden können, benötigen Sie ein Amazon Web Services (AWS)-Konto. Das AWS-Konto ist kostenlos. Sie zahlen nur für die Services und Ressourcen, die Sie wirklich nutzen.

Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen durch.

Anmeldung für ein AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für ein AWS-Konto anmelden, wird ein Root-Benutzer des AWS-Kontos erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Richten Sie die erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen ein.

Für den Zugriff auf die Verwaltung von Amazon DocumentDB-Ressourcen wie Cluster, Instances und Cluster-Parametergruppen sind Anmeldeinformationen erforderlich, die zur Authentifizierung Ihrer Anforderungen verwenden AWS kann. Weitere Informationen finden Sie unter Identity and Access Management für Amazon DocumentDB.

1. Geben Sie in der Suchleiste von IAM AWS Management Consoleein und wählen Sie IAM im daraufhin angezeigten Dropdown-Menü aus.

2. Sobald Sie sich in der IAM-Konsole befinden, wählen Sie im Navigationsbereich Benutzer aus.
3. Wählen Sie Ihren Benutzernamen aus.
4. Klicken Sie auf die Schaltfläche Berechtigungen hinzufügen .
5. Wählen Sie die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.
6. Geben Sie `AmazonDocDBFullAccess` in die Suchleiste ein und wählen Sie sie aus, sobald sie in den Suchergebnissen erscheint.
7. Klicken Sie unten auf die blaue Schaltfläche, auf der Weiter: Überprüfen steht.
8. Klicken Sie unten auf die blaue Schaltfläche mit der Meldung Berechtigungen hinzufügen .

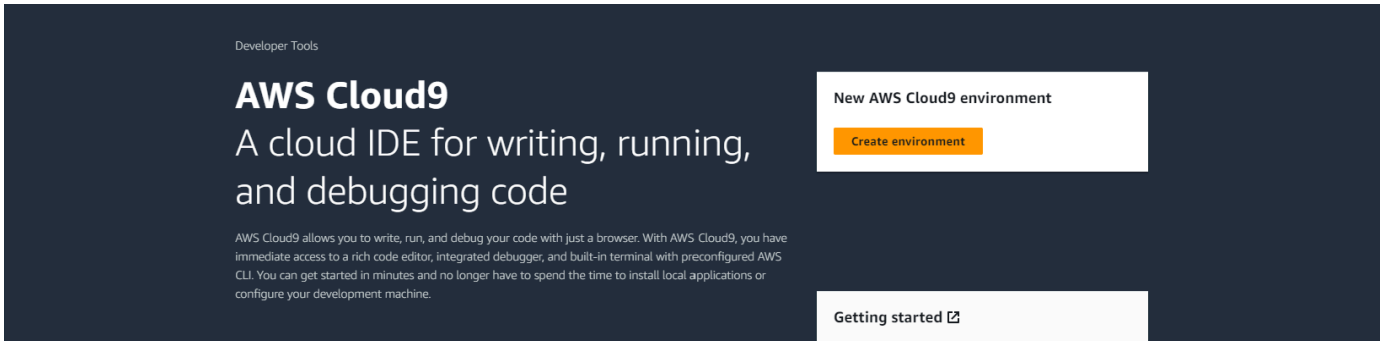
Erstellen einer Amazon Virtual Private Cloud (Amazon VPC)

Dieser Schritt ist nur erforderlich, wenn Sie noch keine Standard-Amazon-VPC haben. Wenn Sie dies nicht tun, führen Sie Schritt 1 von Erste Schritte mit Amazon VPC im Amazon-VPC-Benutzerhandbuch aus. Dies dauert weniger als fünf Minuten.

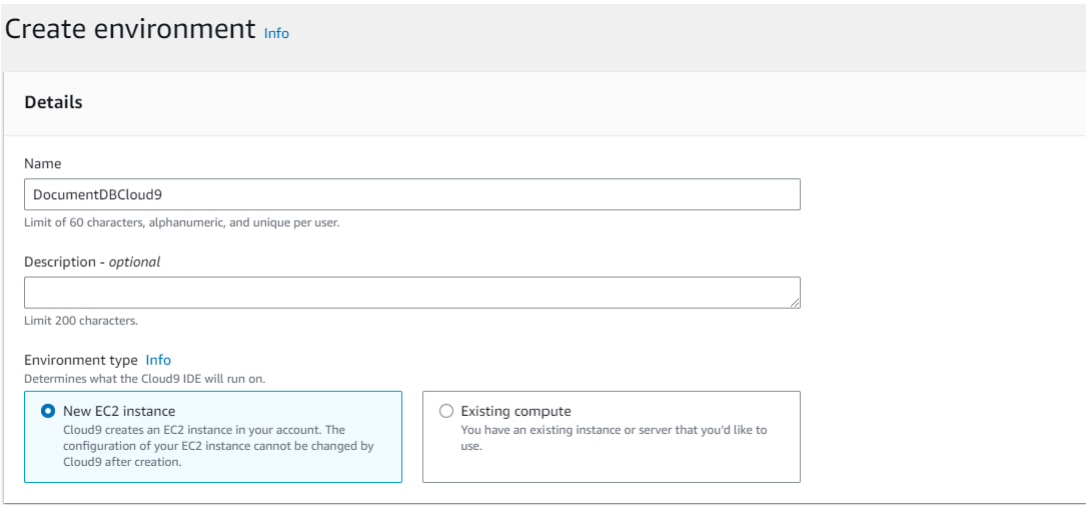
Schritt 1: Erstellen einer AWS Cloud9 Umgebung

AWS Cloud9 bietet ein webbasiertes Terminal, mit dem Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster über die mongo-Shell herstellen und diesen abfragen können.

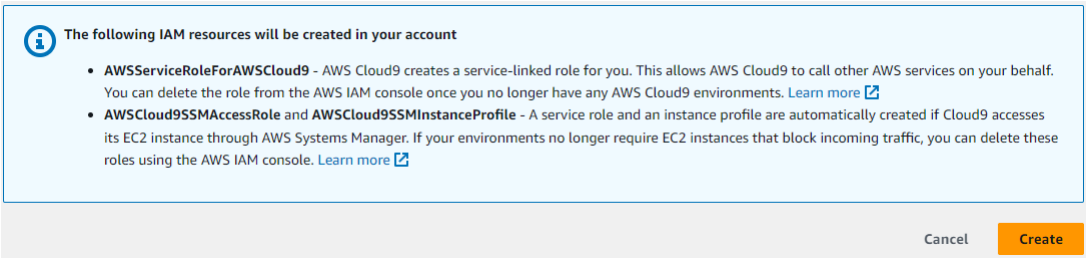
1. Navigieren Sie im AWS Management Console zur -AWS Cloud9Konsole und wählen Sie Umgebung erstellen aus.



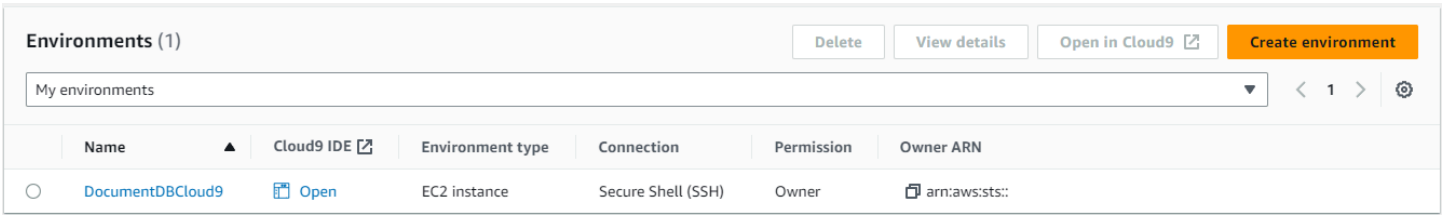
2. Geben Sie im Abschnitt Details des Dialogfelds Umgebung erstellen `DocumentDBCloud9` in das Feld Name ein.



3. Behalten Sie für die Abschnitte Neue EC2-Instance, Netzwerkeinstellungen und Tags die Standardeinstellung unverändert bei und klicken Sie unten auf dem Bildschirm auf Erstellen.



Ihre neue AWS Cloud9 Umgebung wird in der Tabelle Umgebungen angezeigt:

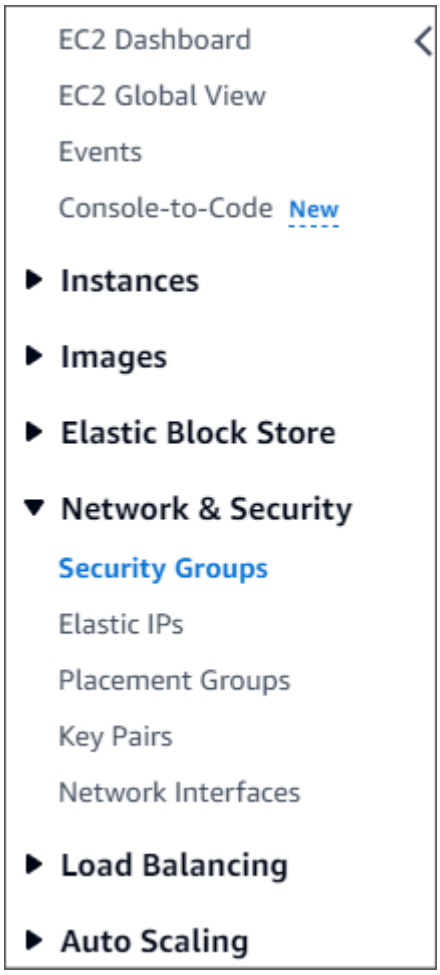


> **Note**
>
> Die Bereitstellung der AWS Cloud9 Umgebung kann bis zu drei Minuten dauern.

Schritt 2: Erstellen einer Sicherheitsgruppe

Mit dieser Sicherheitsgruppe können Sie von Ihrer AWS Cloud9 Umgebung aus eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

1. Wählen Sie in der Amazon EC2-Managementkonsole unter Netzwerk und Sicherheit die Option Sicherheitsgruppen aus.



2. Wählen Sie Sicherheitsgruppe erstellen aus.

3. Im Abschnitt Grundlegende Details:

   a. Geben Sie für Security group name (Name der Sicherheitsgruppe) `demoDocDB` ein.

   b. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein.

   c. Akzeptieren Sie für VPC die Verwendung Ihrer Standard-VPC.

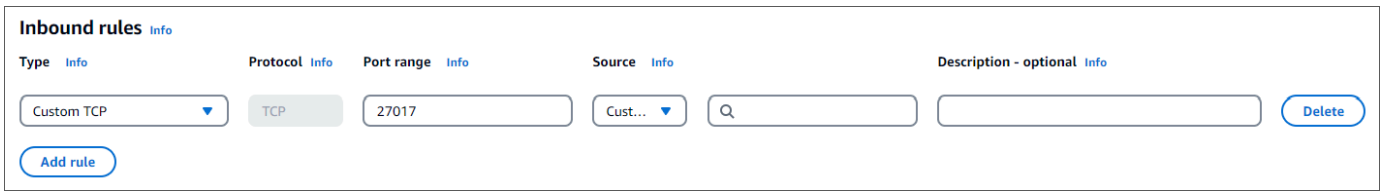
4. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.

   a. Wählen Sie für Type Custom TCP Rule aus.

   b. Geben Sie für Portbereich ein`27017`.

   c. Wählen Sie für Quelle die Sicherheitsgruppe für die soeben erstellte AWS Cloud9 Umgebung aus. Um eine Liste der verfügbaren Sicherheitsgruppen anzuzeigen, geben Sie `cloud9` in das Suchfeld rechts neben dem Feld Quelle ein. Wählen Sie die Sicherheitsgruppe mit dem Namen `aws-cloud9-<environment name>` aus.

   d. Wählen Sie für Ziel die Option Benutzerdefiniert aus. Suchen Sie im Feld daneben nach der Sicherheitsgruppe, die Sie gerade mit dem Namen erstellt habendemoEC2. Möglicherweise müssen Sie Ihren Browser aktualisieren, damit die Amazon EC2-Konsole den demoEC2 Quellnamen automatisch ausfüllen kann.



> **Note**
>
> Port 27017 ist der Standardport für Amazon DocumentDB .

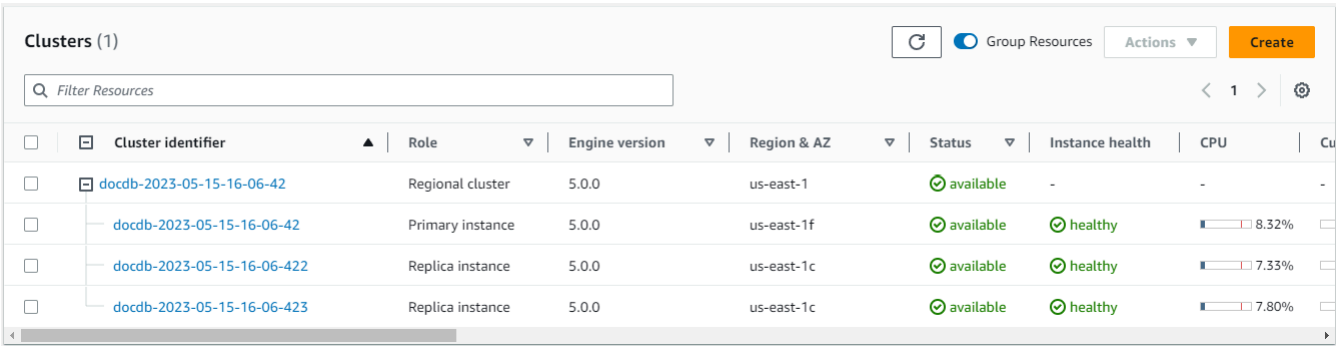
5. Akzeptieren Sie alle anderen Standardwerte und wählen Sie Sicherheitsgruppe erstellen aus.

Schritt 3: Erstellen eines Amazon DocumentDB-Clusters

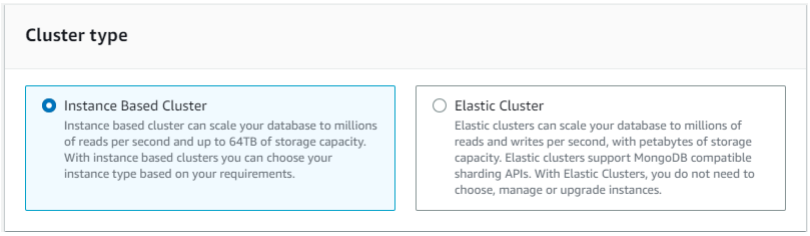
In diesem Schritt erstellen Sie einen Amazon DocumentDB-Cluster mit der Sicherheitsgruppe, die Sie im vorherigen Schritt erstellt haben.

> **Note**
>
> Die Anweisungen in diesem Schritt beziehen sich speziell auf das Erstellen von Instance-basierten Amazon DocumentDB-Clustern. Informationen zum Erstellen elastischer Amazon DocumentDB-Cluster finden Sie unter Erste Schritte mit elastischen Amazon DocumentDB-Clustern.

1. Wählen Sie in der Amazon DocumentDB-Managementkonsole unter Cluster die Option Erstellen aus.

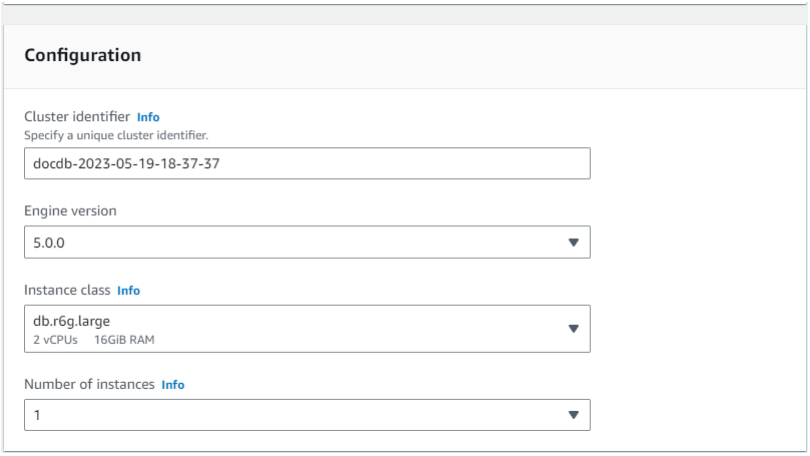


2. Wählen Sie auf der Seite Amazon DocumentDB-Cluster erstellen im Abschnitt Clustertyp die Option Instance-basierte Cluster aus (dies ist die Standardoption).

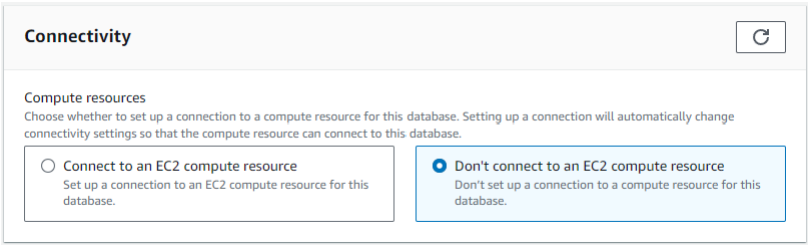


3. Wählen Sie im Abschnitt Konfiguration die Option 1 Instance aus. Die Auswahl einer Instance trägt zur Kostenminimierung bei. Wenn es sich um ein Produktionssystem handelt, empfehlen

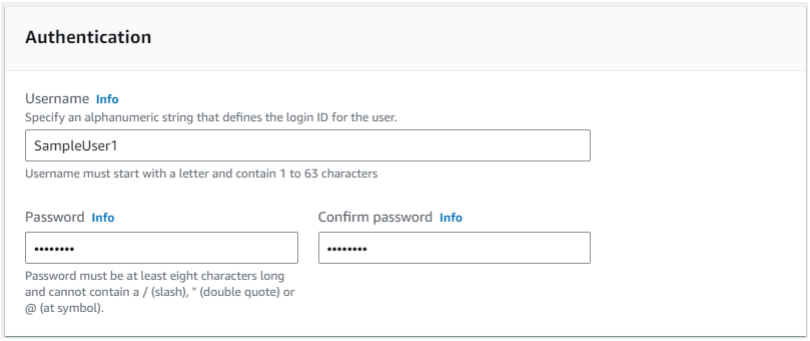
wir Ihnen, drei Instances für hohe Verfügbarkeit bereitzustellen. Sie können die anderen Einstellungen im Abschnitt Konfiguration auf ihre Standardwerte festlegen.



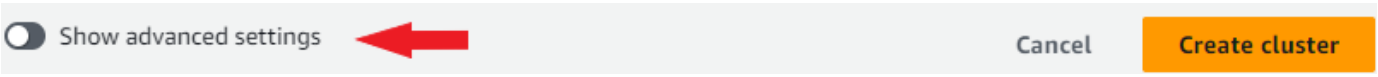
4. Behalten Sie für Konnektivität die Standardeinstellung Keine Verbindung zu einer EC2-Rechenressource herstellt bei.



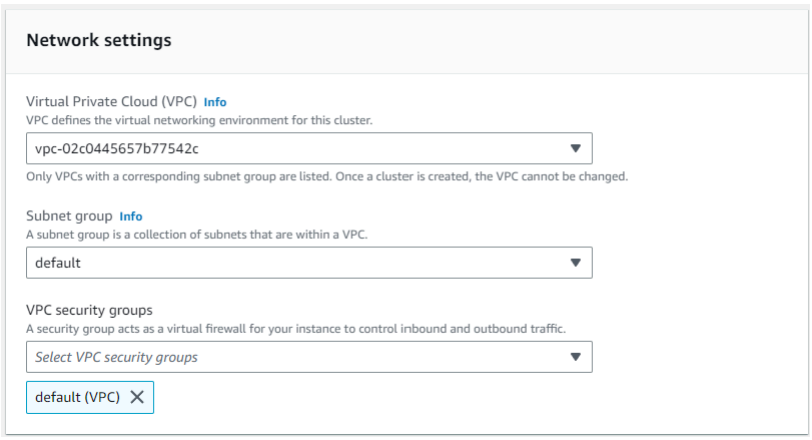
5. Geben Sie im Abschnitt Authentifizierung Anmeldeinformationen ein.



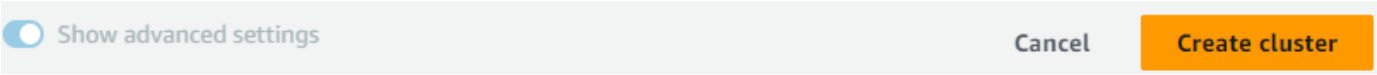
6. Aktivieren Sie Erweiterte Einstellungen anzeigen.



7. Wählen Sie im Abschnitt Netzwerkeinstellungen für VPC-Sicherheitsgruppen die Option demoDocDB (VPC), wenn Sie einen Test- oder Demo-Cluster erstellen. Wenn Sie einen Cluster für ein Produktionssystem erstellen, wählen Sie Standard (VPC) oder wenn Sie eine bestimmte VPC-Sicherheitsgruppe erstellen möchten, finden Sie weitere Informationen unter Sicherheitsgruppen im Amazon Virtual Private Cloud-Benutzerhandbuch.



8. Wählen Sie Cluster erstellen.



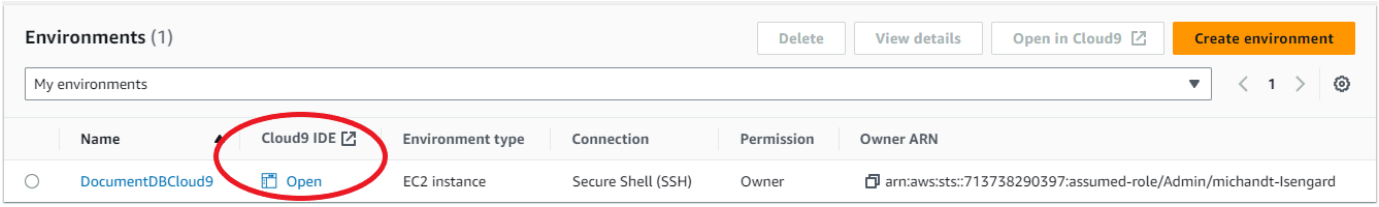
Amazon DocumentDB stellt jetzt Ihren Cluster bereit, was bis zu einigen Minuten dauern kann. Sie können eine Verbindung zu Ihrem Cluster herstellen, wenn sowohl der Cluster- als auch der Instance-Status als angezeigt werden**`available`**.

> **Note**
>
> Weitere Informationen zu Cluster-Statuswerten finden Sie unter Cluster-Statuswerte im Kapitel Überwachen von Amazon DocumentDB.
> Weitere Informationen zu Instance-Statuswerten finden Sie Instance-Statuswerte unter im Kapitel Überwachen von Amazon DocumentDB.

Schritt 4: Installieren der mongo-Shell

Sie installieren jetzt die mongo-Shell in Ihrer AWS Cloud9 Umgebung, die Sie in Schritt 1 erstellt haben. Die mongo-Shell ist ein Befehlszeilendienstprogramm, mit dem Sie Ihren Amazon DocumentDB-Cluster verbinden und abfragen können.

1. Wenn Ihre AWS Cloud9 Umgebung ab Schritt 1 noch offen ist, kehren Sie zu dieser Umgebung zurück und fahren Sie mit Anweisung 3 fort. Wenn Sie die AWS Cloud9 Umgebung verlassen haben, finden Sie in der -AWS Cloud9Managementkonsole unter Umgebungen die Umgebung mit dem Namen DocumentDBCloud9. Wählen Sie in der Spalte Cloud9 IDE die Option Öffnen aus.



2. Erstellen Sie an der Eingabeaufforderung die Repository-Datei mit dem folgenden Befehl:

```
echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-4.0.repo
```

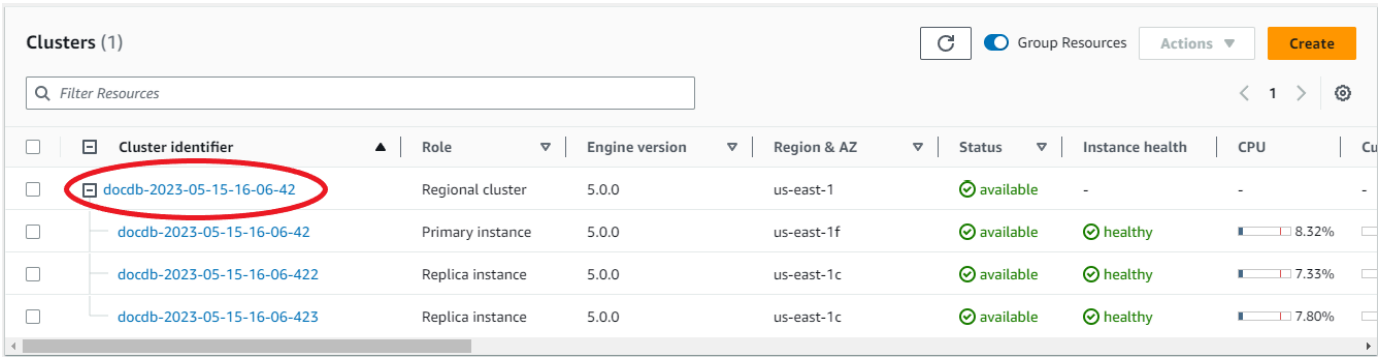
3. Wenn der Vorgang abgeschlossen ist, installieren Sie die mongo-Shell mit dem folgenden Befehl:

```
sudo yum install -y mongodb-org-shell
```

Schritt 5: Herstellen einer Verbindung mit Ihrem Amazon DocumentDB-Cluster

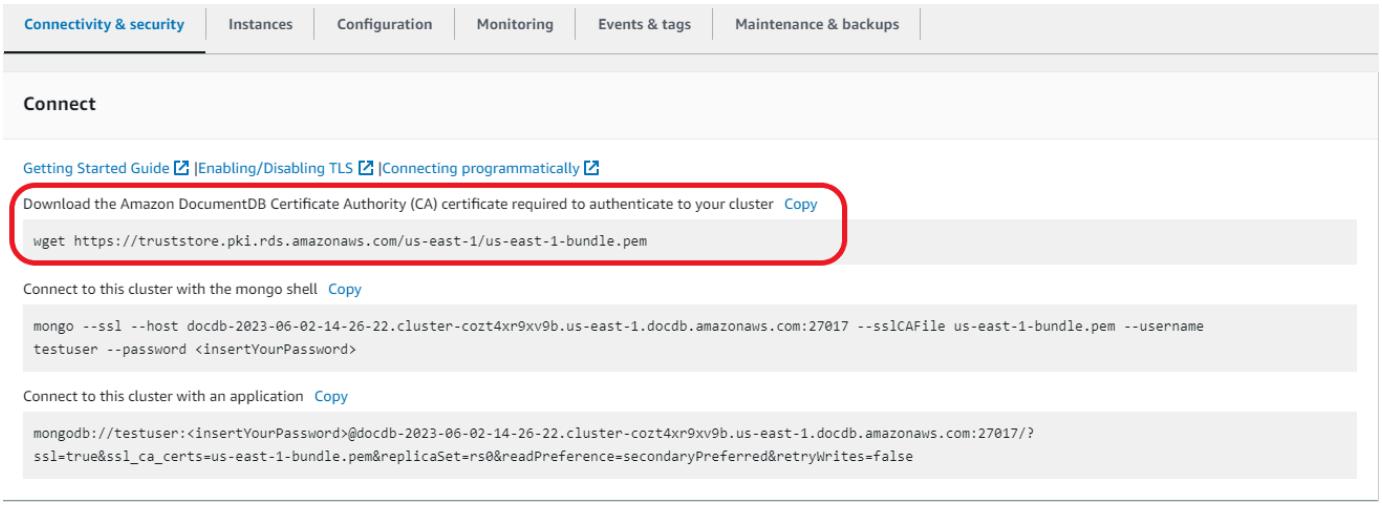
Sie stellen jetzt mithilfe der mongo-Shell, die Sie in Schritt 4 installiert haben, eine Verbindung zu Ihrem Amazon DocumentDB-Cluster her.

1. Suchen Sie in der Amazon DocumentDB-Managementkonsole unter Cluster Ihren Cluster. Wählen Sie den Cluster aus, den Sie erstellt haben, indem Sie auf die Cluster-ID klicken.

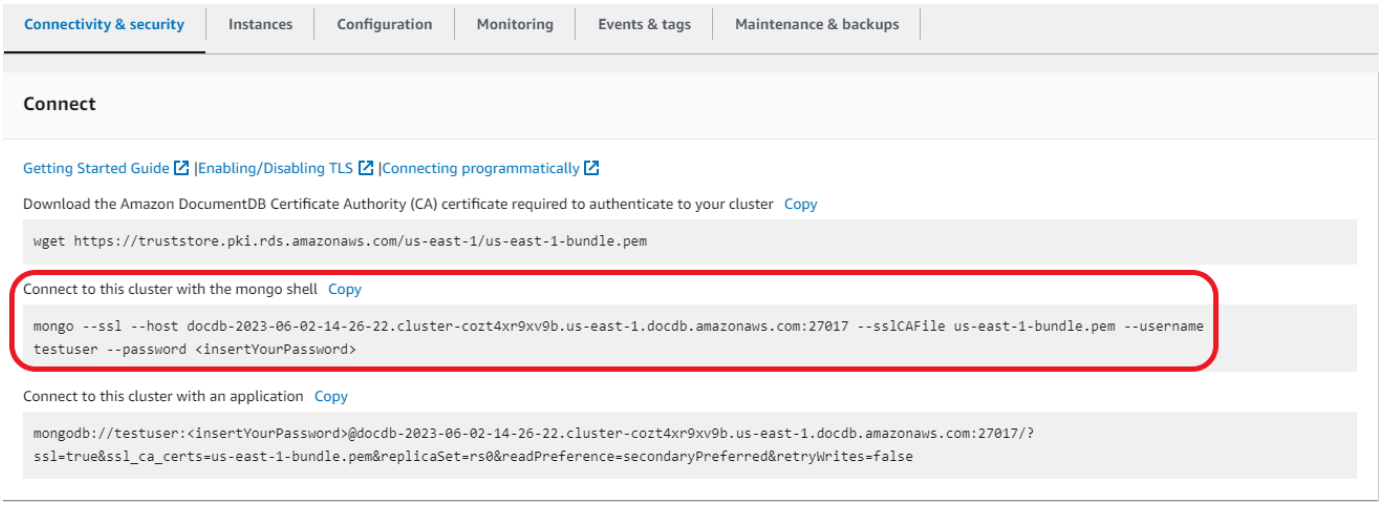


2. E ncryption-in-transit ist auf Amazon DocumentDB standardmäßig aktiviert. Sie können TLS optional deaktivieren. Um das aktuelle Zertifikat herunterzuladen, das für die Authentifizierung bei Ihrem Cluster erforderlich ist, kopieren Sie auf der Registerkarte Konnektivität und Sicherheit im Abschnitt Verbinden unter Herunterladen des Amazon DocumentDB Certificate Authority

(CA)-Zertifikats, das für die Authentifizierung bei Ihrem Cluster erforderlich ist die bereitgestellte Verbindungszeichenfolge. Kehren Sie zu Ihrer AWS Cloud9 Umgebung zurück und fügen Sie die Verbindungszeichenfolge ein.



3. Kehren Sie zu Ihrem Cluster in der Amazon DocumentDB-Konsole zurück und machen Sie die Registerkarte Konnektivität und Sicherheit rückgängig im Abschnitt Verbinden unter Verbindung zu diesem Cluster mit der mongo-Shell herstellen kopieren Sie die angegebene Verbindungszeichenfolge. Lassen Sie das Kopieren aus, `<insertYourPassword>` damit Sie beim Herstellen einer Verbindung von der mongo-Shell zur Eingabe des Passworts aufgefordert werden.



Kehren Sie zu Ihrer AWS Cloud9 Umgebung zurück und fügen Sie die Verbindungszeichenfolge ein.

Wenn Sie Ihr Passwort eingeben und Ihre Eingabeaufforderung `rs0:PRIMARY>` angezeigt wird, sind Sie erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden.

> **Note**
>
> Informationen zur Fehlerbehebung finden Sie unter Fehlerbehebung bei Amazon DocumentDB.

Schritt 6: Einfügen und Abfragen von Daten

Nachdem Sie nun mit Ihrem Cluster verbunden sind, können Sie einige Abfragen ausführen, um sich mit der Verwendung einer Dokumentdatenbank vertraut zu machen.

1. Um ein einzelnes Dokument einzufügen, geben Sie Folgendes ein:

```
db.collection.insert({"hello":"DocumentDB"})
```

2. Sie erhalten die folgende Ausgabe:

```
WriteResult({ "nInserted" : 1 })
```

3. Sie können das Dokument lesen, das Sie mit dem `findOne()` Befehl geschrieben haben (da es nur ein einzelnes Dokument zurückgibt). Geben Sie Folgendes ein:

```
db.collection.findOne()
```

4. Sie erhalten die folgende Ausgabe:

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" : "DocumentDB"
                        }
```

5. Um einige weitere Abfragen durchzuführen, sollten Sie einen Anwendungsfall für Spieleprofile in Betracht ziehen. Fügen Sie zunächst einige Einträge in eine Sammlung mit dem Titel ein`profiles`. Geben Sie Folgendes ein:

```
db.profiles.insertMany([
                    { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
                    { "_id" : 2, "name" : "Frank", "status": "inactive", "level":
 2, "score":9},
                    { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
 "score":87},
```

```
                    { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
 "score":27}
                    ])
```

6. Sie erhalten die folgende Ausgabe:

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. Verwenden Sie den `find()` Befehl , um alle Dokumente in der Profilsammlung zurückzugeben. Geben Sie Folgendes ein:

```
db.profiles.find()
```

8. Sie erhalten eine Ausgabe, die mit den Daten übereinstimmt, die Sie in Schritt 5 eingegeben haben.
9. Verwenden Sie eine Abfrage für ein einzelnes Dokument mithilfe eines Filters. Geben Sie Folgendes ein:

```
db.profiles.find({name: "Katie"})
```

10. Sie sollten diese Ausgabe zurückgeben:

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
                        "score":27}
```

11. Versuchen wir nun, ein Profil zu finden und es mit dem `findAndModify` Befehl zu ändern. Wir vergeben dem Benutzer mit dem folgenden Code weitere zehn Punkte:

```
db.profiles.findAndModify({
                query: { name: "Matt", status: "active"},
                update: { $inc: { score: 10 } }
            })
```

12. Sie erhalten die folgende Ausgabe (beachten Sie, dass sein Wert noch nicht gestiegen ist):

```
{
        "_id" : 1,
        "name" : "Matt",
        "status" : "active",
        "level" : 12,
        "score" : 202
```

```
}
```

13. Mit der folgenden Abfrage können Sie überprüfen, ob sich sein Wert geändert hat:

```
db.profiles.find({name: "Matt"})
```

14. Sie erhalten die folgende Ausgabe:

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12, "score"
                        : 212 }
```

Schritt 7: Erkunden

Herzlichen Glückwunsch! Sie haben das Handbuch „Erste Schritte“ für Amazon DocumentDB erfolgreich abgeschlossen.

Was ist als Nächstes? Erfahren Sie, wie Sie diese Datenbank mit einigen ihrer beliebten Funktionen vollständig nutzen können:

- Verwalten von Amazon DocumentDB
- Skalierung
- Sichern und Wiederherstellen

> **Note**
>
> Für den Cluster, den Sie aus dieser Übung für die ersten Schritte erstellt haben, fallen weiterhin Kosten an, es sei denn, Sie löschen ihn. Anweisungen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Amazon DocumentDB-Schnellstart mit AWS CloudFormation

Dieser Abschnitt enthält Schritte und andere Informationen, die Ihnen den schnellen Einstieg in Amazon DocumentDB (mit MongoDB-Kompatibilität) unter Verwendung von erleichternAWS CloudFormation. Allgemeine Informationen zu Amazon DocumentDB finden Sie unter Was ist Amazon DocumentDB (mit MongoDB-Kompatibilität).

Diese Anweisungen verwenden eine -AWS CloudFormationVorlage, um einen Cluster und Instances in Ihrer standardmäßigen Amazon VPC zu erstellen. Anweisungen zum Erstellen dieser Ressourcen finden Sie unter Erste Schritte mit Amazon DocumentDB.

> **Important**
>
> Der AWS CloudFormationStack, der mit dieser Vorlage erstellt wird, erstellt mehrere Ressourcen, einschließlich Ressourcen in Amazon DocumentDB (z. B. einen Cluster und Instances) und Amazon Elastic Compute Cloud (z. B. eine Subnetzgruppe).
> Einige dieser Ressourcen sind nicht im kostenlosen Kontingent enthalten. Preisinformationen finden Sie unter Amazon DocumentDB – Preise und Amazon EC2 – Preise. Sie können den Stack löschen, wenn Sie ihn nicht mehr benötigen, um Gebühren zu sparen.

Dieser AWS CloudFormationStack dient nur zu Tutorialzwecken. Wenn Sie diese Vorlage für eine Produktionsumgebung verwenden, empfehlen wir Ihnen, strengere IAM-Richtlinien und -Sicherheit zu verwenden. Informationen zum Sichern von Ressourcen finden Sie unter Amazon VPC Security und Amazon EC2 Network and Security.

Themen

Voraussetzungen

Bevor Sie einen Amazon DocumentDB-Cluster erstellen, benötigen Sie Folgendes:

- Eine Standard-Amazon-VPC
- Die erforderlichen IAM-Berechtigungen

Erforderliche IAM-Berechtigungen

Die folgenden Berechtigungen erlauben Ihnen das Erstellen von Ressourcen für den AWS CloudFormation-Stack:

AWS Verwaltete Richtlinien

- `AWSCloudFormationReadOnlyAccess`
- `AmazonDocDBFullAccess`

Zusätzliche IAM-Berechtigungen

Die folgende Richtlinie enthält die zusätzlichen Berechtigungen, die zum Erstellen und Löschen dieses AWS CloudFormation-Stacks erforderlich sind.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "iam:GetSSHPublicKey",
                "iam:ListSSHPublicKeys",
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:PutRolePolicy",
                "iam:CreateInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:GetAccountSummary",
                "iam:ListAccountAliases",
                "iam:GetRole",
                "iam:DeleteRole",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:DeleteRolePolicy",
                "iam:DeleteInstanceProfile",
                "cloudformation:*Stack",
                "ec2:DescribeKeyPairs",
```

```
                "ec2:*Vpc",
                "ec2:DescribeInternetGateways",
                "ec2:*InternetGateway",
                "ec2:createTags",
                "ec2:*VpcAttribute",
                "ec2:DescribeRouteTables",
                "ec2:*RouteTable",
                "ec2:*Subnet",
                "ec2:*SecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:DescribeVpcEndpoints",
                "ec2:*VpcEndpoint",
                "ec2:*SubnetAttribute",
                "ec2:*Route",
                "ec2:*Instances",
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Sid": "iamPassRole",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "rds.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> Die fett dargestellten Berechtigungen in der vorhergehenden Richtlinie sind nur erforderlich, um einen Stack zu löschen: `iam:DeleteRole`, `iam:RemoveRoleFromInstanceProfile`, `iam:DeleteRolePolicy`, `iam:DeleteInstanceProfile` und `ec2:DeleteVpcEndpoints`. Beachten Sie auch, dass `ec2:*Vpc` `ec2:DeleteVpc`-Berechtigungen erteilt.

Amazon-EC2-Schlüsselpaar

Sie müssen über ein Schlüsselpaar (und die PEM-Datei) in der Region verfügen, in der Sie den AWS CloudFormation-Stack erstellen werden. Wenn Sie ein Schlüsselpaar erstellen müssen, finden Sie weitere Informationen unter Erstellen eines Schlüsselpaars mit Amazon EC2 im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Starten eines Amazon DocumentDBAWS CloudFormation-Stacks

In diesem Abschnitt wird beschrieben, wie Sie einen Amazon DocumentDB-AWS CloudFormationStack starten und konfigurieren.

1. Melden Sie sich unter https://console.aws.amazon.com/ bei der AWS Management Console an.
2. In der folgenden Tabelle sind die Amazon DocumentDB-Stack-Vorlagen für jedes aufgeführtAWS-Region. Wählen Sie Stack starten für die aus, in der AWS-Region Sie Ihren Stack starten möchten.

Region	Vorlage anzeigen	In Designer anzeigen	Starten
USA Ost (Ohio)	Vorlage anzeigen	In Designer anzeigen	
USA Ost (Nord-Virginia)	Vorlage anzeigen	In Designer anzeigen	
USA West (Oregon)	Vorlage anzeigen	In Designer anzeigen	
Asien-Pazifik (Mumbai)	Vorlage anzeigen	In Designer anzeigen	
Asien-Pazifik (Seoul)	Vorlage anzeigen	In Designer anzeigen	
Asien-Pazifik (Singapur)	Vorlage anzeigen	In Designer anzeigen	
Asien-Pazifik (Sydney)	Vorlage anzeigen	In Designer anzeigen	

Region	Vorlage anzeigen	In Designer anzeigen	Starten
Asien-Pazifik (Tokio)	Vorlage anzeigen	In Designer anzeigen	Launch Stack
Kanada (Zentral)	Vorlage anzeigen	In Designer anzeigen	Launch Stack
Europa (Frankfurt)	Vorlage anzeigen	In Designer anzeigen	Launch Stack
Europa (Irland)	Vorlage anzeigen	In Designer anzeigen	Launch Stack
Europa (London)	Vorlage anzeigen	In Designer anzeigen	Launch Stack
Europa (Paris)	Vorlage anzeigen	In Designer anzeigen	Launch Stack

3. Stack erstellen – Beschreibt die Amazon DocumentDB-Vorlage, die Sie ausgewählt haben. Jeder Stack basiert auf einer Vorlage – einer JSON- oder YAML-Datei – die Konfiguration über die AWS Ressourcen enthält, die Sie in den Stack aufnehmen möchten. Da Sie sich dafür entschieden haben, einen Stack aus den oben bereitgestellten Vorlagen zu starten, wurde Ihre Vorlage bereits so konfiguriert, dass sie einen Amazon DocumentDB-Stack für die von AWS-Region Ihnen gewählte erstellt.

   Wenn Sie einen -AWS CloudFormationStack starten, ist der Löschschutz für Ihren Amazon DocumentDB-Cluster standardmäßig deaktiviert. Wenn Sie den Löschschutz für Ihren Cluster aktivieren möchten, führen Sie die folgenden Schritte aus. Andernfalls wählen Sie Next (Weiter), um mit dem nächsten Schritt fortzufahren.

   So aktivieren Sie den Löschschutz für Ihren Amazon DocumentDB-Cluster:

   1. Wählen Sie in Designer in der unteren rechten Ecke der Seite Stack erstellen aus.
   2. Ändern Sie die Vorlage mit dem integrierten JSON- und YAML-Editor auf der daraufhin angezeigten AWS CloudFormation-Seite der Konsole. Blättern Sie zum Abschnitt `Resources` und ändern Sie diesen wie folgt, um den `DeletionProtection` einzubinden. Weitere Informationen über die Verwendung des AWS CloudFormation Designer finden Sie unter Was ist AWS CloudFormation Designer?.

      JSON:

```
"Resources": {
    "DBCluster": {
        "Type": "AWS::DocDB::DBCluster",
        "DeletionPolicy": "Delete",
        "Properties": {
            "DBClusterIdentifier": {
                "Ref": "DBClusterName"
            },
            "MasterUsername": {
                "Ref": "MasterUser"
            },
            "MasterUserPassword": {
                "Ref": "MasterPassword"
            },
            "DeletionProtection": "true"
        }
    },
```

YAML:

```
Resources:
  DBCluster:
    Type: 'AWS::DocDB::DBCluster'
    DeletionPolicy: Delete
    Properties:
      DBClusterIdentifier: !Ref DBClusterName
      MasterUsername: !Ref MasterUser
      MasterUserPassword: !Ref MasterPassword
      DeletionProtection: 'true'
```

3. Wählen Sie Create Stack (Stack erstellen)
   ()
   in der oberen linken Ecke der Seite, um Ihre Änderungen zu speichern und einen Stack zu erstellen, bei dem diese Änderungen aktiviert sind.
4. Nachdem Sie Ihre Änderungen gespeichert haben, werden Sie zur Seite Create stack (Stack erstellen) weitergeleitet.
5. Wählen Sie Next (Weiter), um fortzufahren.

4. Stack-Details angeben – Geben Sie den Stack-Namen und die Parameter für Ihre Vorlage ein. Parameter werden in Ihrer Vorlage definiert und ermöglichen Ihnen, benutzerdefinierte Werte einzugeben, wenn Sie einen Stack erstellen oder aktualisieren.
   - Geben Sie unter Stack name (Stack-Name) einen Namen für Ihren Stack ein oder übernehmen Sie den angegebenen Namen. Der Stack-Name kann Buchstaben (A–Z und a–z), Zahlen (0–9) und Bindestriche (–) enthalten.
   - Geben Sie unter Parameter, die folgenden Details ein:
     - DBClusterName – Geben Sie einen Namen für Ihren Amazon DocumentDB-Cluster ein oder akzeptieren Sie den angegebenen Namen.

       Einschränkungen bei der Benennung von Clustern:
       - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
       - Muss mit einem Buchstaben beginnen.
       - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
       - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
     - DBInstanceClass – Wählen Sie aus der Dropdown-Liste die Instance-Klasse für Ihren Amazon DocumentDB-Cluster aus.
     - DBInstanceName – Geben Sie einen Namen für Ihre Amazon DocumentDB-Instance ein oder akzeptieren Sie den angegebenen Namen.

       Einschränkungen für Instance-Benennungen:
       - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
       - Muss mit einem Buchstaben beginnen.
       - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
       - Muss für alle Instances in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
     - MasterPassword – Das Passwort des Datenbankadministratorkontos.
     - MasterUser – Der Benutzername des Datenbankadministratorkontos. Der MasterUser muss mit einem Buchstaben beginnen und darf nur alphanumerische Zeichen enthalten.

   Klicken Sie auf Next (Weiter), um Ihre Änderungen zu speichern und fortzufahren.

5. Konfigurieren von Stack-Optionen – Konfigurieren Sie die Tags, Berechtigungen und zusätzlichen Optionen Ihres Stacks.

   - Tags – Geben Sie Tags (Schlüssel-Wert) an, die auf Ihre Ressourcen in Ihrem Stack angewendet werden sollen. Sie können bis zu 50 eindeutige Tags für jeden Stack hinzufügen.
   - Berechtigungen – Optional. Wählen Sie eine IAM-Rolle aus, um explizit zu definieren, wie Ressourcen im Stack erstellen, ändern oder löschen AWS CloudFormation kann. Wenn Sie keine Rolle auswählen, verwendet AWS CloudFormation auf Ihren Benutzeranmeldeinformationen basierende Berechtigungen. Bevor Sie eine Service-Rolle festlegen, stellen Sie sicher, dass Sie über die Berechtigung zum Weiterleiten (`iam:PassRole`) verfügen. Die `iam:PassRole`-Berechtigung gibt an, welche Rollen Sie verwenden können.

     > **Note**
     >
     > Wenn Sie eine Service-Rolle festlegen, verwendet AWS CloudFormation für alle in diesem Stapel ausgeführten Vorgänge immer diese Rolle. Andere Benutzer, die Berechtigungen zum Ausführen von Vorgängen in diesem Stapel haben, können diese Rolle verwenden, selbst wenn Sie keine Berechtigung zum Weiterleiten haben. Wenn die Rolle Berechtigungen umfasst, die der Benutzer nicht haben sollte, können Sie die Berechtigungen eines Benutzers versehentlich weiterleiten. Stellen Sie sicher, dass die Rolle die geringsten Berechtigungen gewährt.

   - Erweiterte Optionen – Sie können die folgenden erweiterten Optionen festlegen:

     - Stack-Richtlinie – Optional. Legt die Ressourcen, fest die Sie vor unbeabsichtigten Aktualisierungen während einer Stack-Aktualisierung schützen möchten. Standardmäßig können alle Ressourcen bei einer Stack-Aktualisierung aktualisiert werden.

       Sie können die Stack-Richtlinie direkt im JSON-Format eingeben oder eine JSON-Datei, die die Stack-Richtlinie enthält, hochladen. Weitere Informationen finden Sie unter Prevent Updates to Stack Resources (Verhindern von Aktualisierungen der Stack-Ressourcen).
     - Rollback-Konfiguration – optional. Geben Sie CloudWatch Protokollalarme für anAWS CloudFormation, die beim Erstellen und Aktualisieren des Stacks überwacht werden sollen. Wenn die Operation eine Alarmschwelle überschreitet, setzt AWS CloudFormation sie zurück.

- Benachrichtigungsoptionen – Optional. Geben Sie Themen für das Simple Notification System (SNS) an.
- Stack-Erstellungsoptionen – Optional. Sie können die folgenden Optionen angeben:
  - Rollback bei Fehler – Ob der Stack zurückgesetzt werden soll, wenn die Stack-Erstellung fehlschlägt oder nicht.
  - Timeout – Die Anzahl der Minuten vor dem Timeout einer Stack-Erstellung.
  - Beendigungsschutz – Verhindert, dass der Stack versehentlich gelöscht wird.

> **Note**
>
> AWS CloudFormation Der Beendigungsschutz unterscheidet sich vom Amazon DocumentDB-Prinzip des Löschschutzes. Weitere Informationen finden Sie unter Beendigungsschutz und Löschschutz.

Wählen Sie Next (Weiter), um fortzufahren.

6. Überprüfen Sie <stack-name> – Überprüfen Sie Ihre Stack-Vorlage, Details und Konfigurationsoptionen. Sie können auch einen Schnellerstellungs-Link unten auf der Seite öffnen, um Stacks mit denselben Grundkonfigurationen wie diesen zu erstellen.

   - Wählen Sie Create (Erstellen), um den Stack zu erstellen.
   - Alternativ können Sie auch Create change set (Änderungssatz erstellen) wählen. Ein Änderungssatz ist eine Vorschau darauf, wie dieser Stack vor seiner Erstellung konfiguriert wird. Auf diese Weise können Sie verschiedene Konfigurationen prüfen, bevor Sie den Änderungssatz ausführen.

Zugriff auf den Amazon DocumentDB-Cluster

Sobald der AWS CloudFormationStack abgeschlossen ist, können Sie eine Amazon EC2-Instance verwenden, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen. Informationen zum Herstellen einer Verbindung mit einer Amazon EC2-Instance über SSH finden Sie unter Herstellen einer Verbindung mit Ihrer Linux-Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

Nachdem Sie verbunden sind, lesen Sie die folgenden Abschnitte, die Informationen zur Verwendung von Amazon DocumentDB enthalten.

- Schritt 4: Installieren der mongo-Shell
- Löschen eines Amazon DocumentDB-Clusters

Beendigungsschutz und Löschschutz

Es ist eine bewährte Methode für Amazon DocumentDB, den Löschschutz und den Beendigungsschutz zu aktivieren. CloudFormation Der Beendigungsschutz ist eine andere Funktion als das Löschschutzfeature von Amazon DocumentDB.

- Beendigungsschutz – Sie können verhindern, dass ein Stack versehentlich gelöscht wird, indem Sie den Beendigungsschutz für Ihren CloudFormation Stack aktivieren. Wenn ein Benutzer versucht, einen Stack mit aktiviertem Beendigungsschutz zu löschen, schlägt das Löschen fehl und der Stack bleibt unverändert. Der Beendigungsschutz ist standardmäßig deaktiviert, wenn Sie einen Stack mit erstellen CloudFormation. Sie können den Beendigungsschutz aktivieren, wenn Sie den Stack erstellen. Weitere Informationen finden Sie unter Einstellen der AWS CloudFormation-Stack-Optionen.
- Löschschutz – Amazon DocumentDB bietet auch die Möglichkeit, den Löschschutz für einen Cluster zu aktivieren. Wenn ein Benutzer versucht, einen Amazon DocumentDB-Cluster mit aktiviertem Löschschutz zu löschen, schlägt der Löschvorgang fehl und der Cluster bleibt unverändert. Wenn der Löschschutz aktiviert ist, wird vor versehentlichen Löschungen aus Amazon DocumentDB AWS Management Console, AWS CLIund geschützt CloudFormation. Weitere Informationen zum Aktivieren und Deaktivieren des Löschschutzes für einen Amazon DocumentDB-Cluster finden Sie unter Löschschutz.

MongoDB-Kompatibilität

Amazon DocumentDB unterstützt MongoDB-Kompatibilität, einschließlich MongoDB 4.0 und MongoDB 5.0. MongoDB-Kompatibilität bedeutet, dass die überwiegende Mehrheit der Anwendungen, Treiber und Tools, die Sie heute bereits mit Ihren MongoDB-Datenbanken verwenden, mit Amazon DocumentDB verwendet werden können, ohne oder mit wenigen Änderungen. In diesem Abschnitt wird alles beschrieben, was Sie über die Kompatibilität von Amazon DocumentDB mit MongoDB wissen müssen, darunter neue Funktionen und Funktionen, erste Schritte, Migrationspfade und funktionale Unterschiede.

Themen

MongoDB 5.0-Kompatibilität

Themen

Was ist neu in Amazon DocumentDB 5.0

Amazon DocumentDB 5.0 bietet neue Funktionen und Fähigkeiten, darunter Speicherlimits und clientseitige Verschlüsselung auf Feldebene. In der folgenden Zusammenfassung werden einige der wichtigsten Funktionen vorgestellt, die in Amazon DocumentDB 5.0 eingeführt wurden. Eine vollständige Liste der neuen Funktionen finden Sie unterVersionshinweise.

- Das Speicherlimit wurde für alle instanzbasierten Amazon DocumentDB-Cluster und shard-basierten Elastic-Cluster auf 128 TiB erhöht.
- Amazon DocumentDB 5.0 Engine (Version 3.0.775) eingeführt
  - Support für MongoDB 5.0 API-Treiber

- Support für clientseitige Verschlüsselung auf Feldebene (FLE). Sie können jetzt Felder auf der Clientseite verschlüsseln, bevor Sie die Daten in den Amazon DocumentDB-Cluster schreiben. Weitere Informationen finden Sie unter Clientseitige Verschlüsselung auf Feldebene
- Neue Aggregationsoperatoren:, `$dateAdd` `$dateSubtract`
- Unterstützt Indizes mit `$elemMatch` Operator. Das hat zur Folge, `$elemMatch` dass Abfragen zu Indexscans führen.

Amazon DocumentDB unterstützt nicht jede MongoDB 5.0-Funktion. Bei der Entwicklung von Amazon DocumentDB 5.0 haben wir die Funktionen und Fähigkeiten, nach denen uns unsere Kunden am häufigsten gefragt haben, rückwärts gearbeitet. Wir werden weiterhin zusätzliche MongoDB 5.0-Funktionen hinzufügen, je nachdem, was unsere Kunden von uns erwarten. Die aktuelle Liste der unterstützten APIs finden Sie unterUnterstützte MongoDB-APIs, -Operationen und -Datentypen.

Erste Schritte mit Amazon DocumentDB 5.0

Informationen zu den ersten Schritten mit Amazon DocumentDB 5.0 finden Sie im Handbuch Erste Schritte. Sie können einen neuen Amazon DocumentDB 5.0-Cluster mit dem AWS Management Console oder dem AWS SDK,AWS CLI, oder AWS CloudFormation erstellen. Wenn Sie eine Verbindung zu Amazon DocumentDB herstellen, müssen Sie einen MongoDB-Treiber oder ein MongoDB-Tool verwenden, das mit MongoDB 5.0 oder höher kompatibel ist.

> **Note**
>
> Wenn Sie das AWS SDK oder AWS CloudFormation verwendenAWS CLI, ist die Engine-Version standardmäßig 5.0.0. Sie müssen den Parameter explizit angeben`engineVersion = 4.0.0`, um einen neuen Amazon DocumentDB 4.0-Cluster oder einen neuen Amazon DocumentDB 3.6-Cluster `engineVersion = 3.6.0` zu erstellen. Für einen bestimmten Amazon DocumentDB-Cluster können Sie die Cluster-Version ermitteln, indem Sie AWS CLI den aufrufen `describe-db-clusters` oder die Amazon DocumentDB-Managementkonsole verwenden, um die Engine-Versionsnummer für einen bestimmten Cluster anzuzeigen.

Amazon DocumentDB 5.0 unterstützt Amazon EC2 Graviton2-Prozessoren wie `r6g` `t4.medium` Instance-Typen für Ihre Cluster und ist in allen unterstützten Regionen verfügbar. Weitere

Informationen zur Preisgestaltung finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) — Preise.

Upgrade oder Migration zu Amazon DocumentDB 4.0

Sie können von MongoDB 3.6 oder MongoDB 4.0 zu Amazon DocumentDB 5.0 migrieren, indem Sie Dienstprogramme wie `mongodump`,`mongorestore`, `mongoimport` und verwenden. AWS DMS`mongoexport` Anweisungen zur Migration finden Sie unter. Aktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service

Funktionsunterschiede

Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und 5.0

Mit der Veröffentlichung von Amazon DocumentDB 5.0 gibt es funktionale Unterschiede zwischen Amazon DocumentDB 3.6 und Amazon DocumentDB 4.0:

- Die integrierte Backup-Rolle unterstützt jetzt. `serverStatus` Aktion — Entwickler und Anwendungen mit Backup-Rolle können Statistiken über den Status des Amazon DocumentDB-Clusters sammeln.
- Das `SecondaryDelaySecs` Feld ersetzt `slaveDelay` in der `replSetGetConfig` Ausgabe.
- Der hello Befehl ersetzt `isMaster` — hello gibt ein Dokument zurück, das die Rolle eines Amazon DocumentDB-Clusters beschreibt.
- Amazon DocumentDB 5.0 unterstützt jetzt Indexscans mit dem `$elemMatch` Operator in der ersten Verschachtelungsebene. Indexscans werden unterstützt, wenn der Filter „Nur Abfrage“ eine Filterebene hat, aber nicht unterstützt, wenn eine verschachtelte `$elemMatch` Abfrage enthalten ist. `$elemMatch`

  Wenn Sie beispielsweise in Amazon DocumentDB 5.0 den `$elemMatch` Operator in die verschachtelte Ebene aufnehmen, gibt er keinen Wert zurück, wie dies in Amazon DocumentDB 4.0 der Fall ist:

```
db.foo.insert(
[
    {a: {b: 5}},
    {a: {b: [5]}},
    {a: {b: [3, 7]}},
    {a: [{b: 5}]},
```

```
    {a: [{b: 3}, {b: 7}]},
    {a: [{b: [5]}]},
    {a: [{b: [3, 7]}]},
    {a: [[{b: 5}]]},
    {a: [[{b: 3}, {b: 7}]]},
    {a: [[{b: [5]}]]},
    {a: [[{b: [3, 7]}]]}
]);

// DocumentDB 5.0
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }

// DocumentDB 4.0
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }
```

- Die Projektion „$“ in Amazon DocumentDB 4.0 gibt alle Dokumente mit allen Feldern zurück. Bei Amazon DocumentDB 5.0 gibt der find Befehl mit einer „$“ -Projektion Dokumente zurück, die dem Abfrageparameter entsprechen und nur das Feld enthalten, das der Projektion „$“ entspricht.
- In Amazon DocumentDB 5.0 geben die find Befehle mit `$regex` und `$options` Abfrageparametern einen Fehler zurück: „Optionen können nicht in beiden `$regex` und festgelegt werden`$options`“.
- In Amazon DocumentDB 5.0 wird `$indexOfCP` jetzt „-1" zurückgegeben, wenn:
  - die Teilzeichenfolge wurde im Zeichenkettenausdruck nicht gefunden, oder
  - Start ist eine Zahl größer als Ende, oder
  - start ist eine Zahl, die größer als die Bytelänge der Zeichenfolge ist.
- Gibt in Amazon DocumentDB 4.0 „0" `$indexOfCP` zurück, wenn die Startposition eine Zahl ist, die größer als das Ende oder die Bytelänge der Zeichenfolge ist.
- Mit Amazon DocumentDB 5.0 geben `_id fields` Projektionsoperationen beispielsweise Dokumente zurück`{"_id.nestedField" : 1}`, die nur das projizierte Feld enthalten. In Amazon DocumentDB 4.0 hingegen filtern Befehle zur verschachtelten Feldprojektion kein Dokument heraus.

MongoDB 4.0-Kompatibilität

Themen

Funktionen von Amazon DocumentDB 4.0

Amazon DocumentDB 4.0 führte viele neue Funktionen und Fähigkeiten ein, darunter ACID-Transaktionen und Verbesserungen bei Change-Streams. Die folgende Zusammenfassung zeigt einige der wichtigsten Funktionen, die in Amazon DocumentDB 4.0 eingeführt wurden. Eine vollständige Liste der Funktionen finden Sie unterVersionshinweise.

- ACID-Transaktionen: Amazon DocumentDB unterstützt jetzt die Möglichkeit, Transaktionen über mehrere Dokumente, Kontoauszüge, Sammlungen und Datenbanken hinweg durchzuführen. Transaktionen vereinfachen die Anwendungsentwicklung, indem sie es Ihnen ermöglichen, atomare, konsistente, isolierte und dauerhafte Operationen (ACID) für ein oder mehrere Dokumente innerhalb eines Amazon DocumentDB-Clusters durchzuführen. Weitere Informationen finden Sie unter Transaktionen.
- Streams ändern: Sie haben jetzt die Möglichkeit, einen Change-Stream auf Cluster-Ebene (`client.watch()`oder`mongo.watch()`) und in der Datenbank (`db.watch()`) zu öffnen, Sie können einen Cursor angeben, `startAtOperationTime` um einen Change-Stream zu öffnen, und schließlich können Sie Ihren Change-Stream-Aufbewahrungszeitraum jetzt auf 7 Tage (zuvor 24 Stunden) verlängern. Weitere Informationen finden Sie unter Change Streams mit Amazon DocumentDB verwenden.
- AWS Database Migration Service(AWS DMS): Sie können jetzt Ihre MongoDB 4.0-Workloads AWS DMS zu Amazon DocumentDB migrieren. AWS DMSunterstützt jetzt eine MongoDB 4.0-Quelle, ein Amazon DocumentDB 4.0-Ziel und eine Amazon DocumentDB 3.6-Quelle für die Durchführung von Upgrades zwischen Amazon DocumentDB 3.6 und 4.0. Weitere Informationen finden Sie in der AWS DMSDokumentation.
- Leistung und Indizierung: Sie können jetzt einen Index verwenden`$lookup`, Abfragen mit einer Projektion suchen, die ein Feld oder ein Feld enthalten, und das `_id` Feld kann direkt aus dem Index bereitgestellt werden, ohne dass aus der Sammlung gelesen werden muss (abgedeckte Abfrage), die Möglichkeit`findAndModify`, `hint()` mit, Leistungsoptimierungen für `$addToSet` und Verbesserungen zur Reduzierung der Gesamtindexgröße. Weitere Informationen finden Sie unter Versionshinweise.

- Operatoren: Amazon DocumentDB 4.0 unterstützt jetzt eine Reihe neuer Aggregationsoperatoren:`$ifNull`,,`$replaceRoot`,, `$setIsSubset``$setInstersection`,`$setUnion`. `$setEquals` Alle MongoDB-APIs, -Operationen und -Datentypen, die wir unterstützen, finden Sie unterUnterstützte MongoDB-APIs, -Operationen und -Datentypen.
- Rollenbasierte Zugriffskontrolle (RBAC): Bei `ListDatabase` Befehlen sowohl als auch `ListCollection` können Sie jetzt optional die `authorizedDatabases` Parameter `authorizedCollections` und verwenden, damit Benutzer die Sammlungen und Datenbanken auflisten können, auf die sie zugreifen dürfen, ohne dass die `listCollections` `listDatabase` Rollen bzw. Sie haben auch die Möglichkeit, Ihre eigenen Cursor zu beenden, ohne die Rolle zu benötigen. `KillCursor`

Amazon DocumentDB unterstützt nicht alle MongoDB 4.0-Funktionen. Bei der Entwicklung von Amazon DocumentDB 4.0 haben wir die Funktionen und Fähigkeiten, nach denen uns unsere Kunden am häufigsten gefragt haben, rückwärts gearbeitet. Wir werden weiterhin zusätzliche MongoDB 4.0-Funktionen hinzufügen, je nachdem, was unsere Kunden von uns erwarten. Beispielsweise unterstützt Amazon DocumentDB 4.0 derzeit nicht die Typkonvertierungsoperatoren oder Zeichenkettenoperatoren, die in MongoDB 4.0 eingeführt wurden. Die aktuelle Liste der unterstützten APIs finden Sie unter. Unterstützte MongoDB-APIs, -Operationen und -Datentypen

Erste Schritte mit Amazon DocumentDB 4.0

Informationen zu den ersten Schritten mit Amazon DocumentDB 4.0 finden Sie im Handbuch Erste Schritte. Sie können einen neuen Amazon DocumentDB 4.0-Cluster mit dem AWS Management Console oder dem AWS SDK,AWS CLI, oder AWS CloudFormation erstellen. Wenn Sie eine Verbindung zu Amazon DocumentDB herstellen, müssen Sie einen MongoDB-Treiber oder ein MongoDB-Tool verwenden, das mit MongoDB 4.0 oder höher kompatibel ist.

> **Note**
>
> Wenn Sie das AWS SDK oder AWS CloudFormation verwendenAWS CLI, ist die Engine-Version standardmäßig 5.0.0. Sie müssen den Parameter explizit angeben`engineVersion = 4.0.0`, um einen neuen Amazon DocumentDB 4.0-Cluster oder einen neuen Amazon DocumentDB 3.6-Cluster `engineVersion = 3.6.0` zu erstellen. Für einen bestimmten Amazon DocumentDB-Cluster können Sie die Cluster-Version ermitteln, indem Sie AWS CLI den aufrufen `describe-db-clusters` oder die Amazon DocumentDB-

Managementkonsole verwenden, um die Engine-Versionsnummer für einen bestimmten Cluster anzuzeigen.

Amazon DocumentDB 4.0 unterstützt `r5r6g`,`t3.medium`, und `t4g.medium` Instance-Typen für Ihre Cluster und ist in allen unterstützten Regionen verfügbar. Für die Nutzung von Amazon DocumentDB 4.0 fallen keine zusätzlichen Kosten an. Weitere Informationen zur Preisgestaltung finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) — Preise.

Upgrade oder Migration zu Amazon DocumentDB 4.0

Sie können von MongoDB 3.6 oder MongoDB 4.0 zu Amazon DocumentDB 4.0 migrieren, indem Sie Dienstprogramme wie `mongodump`,`mongorestore`, `mongoimport` und verwenden. AWS DMS`mongoexport` Ebenso können Sie dieselben Tools für ein Upgrade von Amazon DocumentDB 3.6 auf Amazon DocumentDB 4.0 verwenden. Anweisungen zur Migration finden Sie unter. Aktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service

Funktionsunterschiede

Funktionale Unterschiede zwischen Amazon DocumentDB 3.6 und 4.0

Mit der Veröffentlichung von Amazon DocumentDB 4.0 gibt es funktionale Unterschiede zwischen Amazon DocumentDB 3.6 und Amazon DocumentDB 4.0:

- Projektion für verschachtelte Dokumente: Amazon DocumentDB 3.6 berücksichtigt bei der Anwendung einer Projektion das erste Feld in einem verschachtelten Dokument. Amazon DocumentDB 4.0 analysiert jedoch Unterdokumente und wendet die Projektion auch auf jedes Unterdokument an. Beispiel: Wenn die Projektion ist`"a.b.c": 1`, dann ist das Verhalten in beiden Versionen identisch. Wenn die Projektion jedoch `{a:{b:{c:1}}}` so ist, wendet Amazon DocumentDB 3.6 die Projektion nur auf 'a' und nicht auf 'b' oder 'c' an.
- Verhalten für**`minKey`**, **`maxKey`**: In Amazon DocumentDB 4.0 `{x:{$gt:MaxKey}}` gibt das Verhalten für nichts und für alles `{x:{$lt:MaxKey}}` zurück.
- Unterschiede beim Vergleich von Dokumenten: Der Vergleich von numerischen Werten verschiedener Typen (Double, Int, Long) in Unterdokumenten (z. B. **`b`** in**`{"_id" :1, "a" : {"b":1}}`**) bietet jetzt eine konsistente Ausgabe für alle numerischen Datentypen und für jede Ebene eines Dokuments.

Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und MongoDB 4.0

Im Folgenden sind die funktionalen Unterschiede zwischen Amazon DocumentDB 4.0 und MongoDB 4.0 aufgeführt.

- Suche mit leerem Schlüssel im Pfad: Wenn eine Sammlung ein Dokument mit leerem Schlüssel innerhalb des Arrays enthält (z. B.`{"x" : [ { "" : 10 }, { "b" : 20 } ]}`) und wenn der in der Abfrage verwendete Schlüssel mit einer leeren Zeichenfolge endet (z. B.`x.`), gibt Amazon DocumentDB dieses Dokument zurück, da es alle Dokumente im Array durchläuft, während MongoDB dieses Dokument nicht zurückgibt.
- **`$setOnInsert`**zusammen mit **`$`** im Pfad: Der Feldoperator `$setOnInsert` funktioniert nicht in Kombination mit $ im Pfad in Amazon DocumentDB, was auch mit MongoDB 4.0 konsistent ist.

Transaktionen

Amazon DocumentDB (mit MongoDB-Kompatibilität) unterstützt jetzt MongoDB 4.0-Kompatibilität, einschließlich Transaktionen. Sie können Transaktionen in mehreren Dokumenten, Kontoauszügen, Sammlungen und Datenbanken durchführen. Transaktionen vereinfachen die Anwendungsentwicklung, da Sie atomare, konsistente, isolierte und dauerhafte (ACID) -Operationen (ACID) für ein oder mehrere Dokumente innerhalb eines Amazon DocumentDB-Clusters ausführen können. Zu den häufigsten Anwendungsfällen für Transaktionen gehören die Finanzabwicklung, die Erfüllung und Verwaltung von Aufträgen sowie die Entwicklung von Mehrspieler-Spielen.

Es fallen keine zusätzlichen Kosten für Transaktionen an. Sie zahlen nur für die Lese- und Schreib-IOs, die Sie im Rahmen der Transaktionen verwenden.

Themen

Voraussetzungen

Zur Verwendung der Transaktionsfunktion müssen Sie die folgenden Anforderungen erfüllen:

- Sie müssen die Amazon DocumentDB 4.0-Engine verwenden.
- Sie müssen einen Treiber verwenden, der mit MongoDB 4.0 oder höher kompatibel ist.

Bewährte Methoden

Im Folgenden finden Sie einige bewährte Methoden, damit Sie Transaktionen mit Amazon DocumentDB optimal nutzen können.

- Bestätigen oder brechen Sie die Transaktion immer ab, nachdem sie abgeschlossen ist. Wenn eine Transaktion in einem unvollständigen Zustand belassen wird, werden Datenbankressourcen gebunden und es kann zu Schreibkonflikten kommen.
- Es wird empfohlen, Transaktionen auf die geringste Anzahl von benötigten Befehlen zu beschränken. Wenn Sie Transaktionen mit mehreren Kontoauszügen haben, die in mehrere kleinere Transaktionen aufgeteilt werden können, ist es ratsam, dies zu tun, um die Wahrscheinlichkeit eines Timeouts zu verringern. Versuchen Sie immer, kurze Transaktionen zu erstellen, keine lang andauernden Lesevorgänge.

Einschränkungen

- Amazon DocumentDB unterstützt keine Cursor innerhalb einer Transaktion.
- Amazon DocumentDB kann in einer Transaktion keine neuen Sammlungen erstellen und nicht vorhandene Sammlungen nicht abfragen/aktualisieren.
- Schreibsperren auf Dokumentenebene unterliegen einem Timeout von 1 Minute, das vom Benutzer nicht konfigurierbar ist.
- Befehle für wiederholbare Schreibvorgänge, wiederholbares Commit und wiederholbares Abbrechen werden in Amazon DocumentDB nicht unterstützt. Ausnahme: Wenn Sie Mongo Shell verwenden, fügen Sie den`retryWrites=false` Befehl in keine Codezeichenfolge ein. Standardmäßig werden wiederholbare Schreibvorgänge deaktiviert. Das`retryWrites=false` Einbeziehen kann zu Fehlern bei normalen Lesebefehlen führen.
- Jede Amazon DocumentDB DocumentDB-Instance hat eine Obergrenze für die Anzahl gleichzeitiger Transaktionen, die auf der Instance gleichzeitig geöffnet sind. Die Grenzwerte finden Sie unterInstance-Limits.
- Für eine bestimmte Transaktion muss die Größe des Transaktionsprotokolls weniger als 32 MB betragen.
- Amazon DocumentDB unterstützt zwar Transaktionen`count()` innerhalb einer Transaktion, aber nicht alle Treiber unterstützen diese Funktion. Eine Alternative ist die Verwendung der`countDocuments()` API, die die Zählabfrage in eine Aggregationsabfrage auf der Clientseite übersetzt.

- Transaktionen haben ein Ausführungslimit von einer Minute und Sitzungen haben ein Timeout von 30 Minuten. Wenn eine Transaktion ein Timeout erreicht, wird sie abgebrochen, und alle nachfolgenden Befehle, die innerhalb der Sitzung für die bestehende Transaktion ausgegeben werden, führen zu folgendem Fehler:

```
WriteCommandError({
"ok" : 0,
"operationTime" : Timestamp(1603491424, 627726),
"code" : 251,
"errmsg" : "Given transaction number 0 does not match any in-progress transactions."
})
```

Überwachung und Diagnose

Mit der Unterstützung von Transaktionen in Amazon DocumentDB 4.0 wurden zusätzliche CloudWatch Metriken hinzugefügt, mit denen Sie Ihre Transaktionen überwachen können.

Neue CloudWatch Metriken

- `DatabaseTransactions`: Die Anzahl der offenen Transaktionen, die in einem Zeitraum von einer Minute abgeschlossen wurden.
- `DatabaseTransactionsAborted`: Die Anzahl der abgebrochenen Transaktionen innerhalb eines Zeitraums von einer Minute.
- `DatabaseTransactionsMax`: Die maximale Anzahl offener Transaktionen in einem Zeitraum von einer Minute.
- `TransactionsAborted`: Die Anzahl der Transaktionen, die auf einer Instance in einem Zeitraum von einer Minute abgebrochen wurden.
- `TransactionsCommitted`: Die Anzahl der Transaktionen, die in einem Zeitraum von einer Minute auf einer Instance durchgeführt wurden.
- `TransactionsOpen`: Die Anzahl der offenen Transaktionen für eine Instance innerhalb eines Zeitraums von einer Minute.
- `TransactionsOpenMax`: Die maximale Anzahl von Transaktionen, die in einem Zeitraum von einer Minute auf einer Instance geöffnet wurden.
- `TransactionsStarted`: Die Anzahl der Transaktionen, die in einem Zeitraum von einer Minute auf einer Instance gestartet wurden.

> **Note**
>
> Weitere CloudWatch Metriken für Amazon DocumentDB finden Sie unterÜberwachen von Amazon DocumentDB mit CloudWatch.

Zusätzlich wurden sowohl`currentOplsid` neue Felder als auch ein neuer Status für „`idle transaction`“ und`serverStatus` Transaktionen hinzugefügt:`currentActive`,,`currentInactive`,`currentOpen`,`totalAborted`,`totalCommitted`, und`totalStarted`.`transactionThreadId`

Transaktionsisolierungsstufe

Wenn Sie eine Transaktion starten, haben Sie die Möglichkeit,`readConcern` sowohl die als auch anzugeben,`writeConcern` wie im folgenden Beispiel gezeigt:

```
mySession.startTransaction({readConcern: {level: 'snapshot'}, writeConcern:
{w: 'majority'}});
```

Denn`readConcern` Amazon DocumentDB unterstützt standardmäßig die Snapshot-Isolierung. Wenn lokale, verfügbare oder mehrheitliche Werte angegeben sind, aktualisiert Amazon DocumentDB die`readConcern` Ebene auf Snapshot.`readConcern` Amazon DocumentDB unterstützt das Linearisierbare nicht,`readConcern` und die Angabe eines solchen Leseproblems führt zu einem Fehler.

Amazon DocumentDB unterstützt standardmäßig die Mehrheit`writeConcern`, und ein Schreibquorum wird erreicht, wenn vier Kopien der Daten auf drei AZs gespeichert werden. Wenn ein niedrigerer`writeConcern` Wert angegeben wird, führt Amazon DocumentDB ein Upgrade`writeConcern` auf eine Mehrheit durch. Außerdem werden alle Amazon DocumentDB DocumentDB-Schreibvorgänge protokolliert und das Journaling kann nicht deaktiviert werden.

Anwendungsfälle

In diesem Abschnitt gehen wir auf zwei Anwendungsfälle für Transaktionen ein: mehrere Kontoauszüge und mehrere Inkasso.

Transaktionen mit mehreren Kontoauszügen

Amazon DocumentDB-Transaktionen sind mehrere Anweisungen, d. h. Sie können eine Transaktion schreiben, die mehrere Anweisungen umfasst, mit einem expliziten Commit oder Rollback. Sie können`findAndModify` Aktionen als einzelne atomare Operation gruppieren`insert`.`update``delete`

Ein häufiger Anwendungsfall für Transaktionen mit mehreren Kontoauszügen ist eine Debit-/Kredittransaktion. Zum Beispiel: Du schuldest einem Freund Geld für Kleidung. Daher musst du 500$ von deinem Konto abbuchen (abheben) und 500$ (Einzahlung) auf das Konto deines Freundes gutschreiben. Um diesen Vorgang durchzuführen, führen Sie sowohl die Schulden- als auch die Kredittransaktionen innerhalb einer einzigen Transaktion durch, um die Atomität zu gewährleisten. Auf diese Weise wird verhindert, dass 500$ von Ihrem Konto abgebucht, aber nicht dem Konto Ihres Freundes gutgeschrieben werden. So würde dieser Anwendungsfall aussehen:

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success Scenario***
// Setup bank account for Alice and Bob. Each have $1000 in their account

var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
```

```
accountColl.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;

session.commitTransaction();

accountColl.find();

// *** Transfer $500 from Alice to Bob inside a transaction: Failure Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;

session.abortTransaction();
```

Transaktionen mit mehreren Inkassogeschäften

Bei unseren Transaktionen handelt es sich auch um Mehrfacheinkäufe, was bedeutet, dass sie verwendet werden können, um mehrere Vorgänge innerhalb einer einzigen Transaktion und über mehrere Inkasso hinweg durchzuführen. Dies bietet eine konsistente Ansicht der Daten und gewährleistet die Integrität Ihrer Daten. Wenn Sie die Befehle als einzelne ausführen<>, all-or-nothing handelt es sich bei den Transaktionen um Ausführungsvorgänge. Sie werden entweder alle erfolgreich sein oder alle fehlschlagen.

Hier ist ein Beispiel für Transaktionen mit mehreren Inkasso, bei denen dasselbe Szenario und dieselben Daten aus dem Beispiel für Transaktionen mit mehreren Kontoauszügen verwendet werden.

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
var amountToTransfer = 500;
var collectionName = "account";

var session = db.getMongo().startSession({causalConsistency: false});
var accountCollInBankA = session.getDatabase("bankA")[collectionName];
var accountCollInBankB = session.getDatabase("bankB")[collectionName];

accountCollInBankA.drop();
accountCollInBankB.drop();

accountCollInBankA.insert({name: "Alice", balance: 1000});
accountCollInBankB.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountCollInBankA.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountCollInBankB.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;

session.commitTransaction();

accountCollInBankA.find(); // Alice holds $500 in bankA
accountCollInBankB.find(); // Bob holds $1500 in bankB

// *** Transfer $500 from Alice to Bob inside a transaction: Failure Scenario***

// Setup bank account for Alice and Bob. Each have $1000 in their account
```

```
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var accountCollInBankA = session.getDatabase("bankA")[collectionName];
var accountCollInBankB = session.getDatabase("bankB")[collectionName];

accountCollInBankA.drop();
accountCollInBankB.drop();

accountCollInBankA.insert({name: "Alice", balance: 1000});
accountCollInBankB.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;
var newAliceBalance = aliceBalance - amountToTransfer;
accountCollInBankA.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountCollInBankA.find({"name": "Alice"}).next().balance;

// add $500 to Bob's account
var bobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountCollInBankB.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountCollInBankB.find({"name": "Bob"}).next().balance;

session.abortTransaction();

accountCollInBankA.find(); // Alice holds $1000 in bankA
accountCollInBankB.find(); // Bob holds $1000 in bankB
```

Transaktions-API-Beispiele für die Callback-API

Die Callback-API ist nur für Treiber ab 4.2 verfügbar.

Javascript

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Javascript verwenden.

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success ***
```

```
// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert(aliceBalance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert.eq(newAliceBalance, findAliceBalance);

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;
assert.eq(newBobBalance, findBobBalance);

session.commitTransaction();

accountColl.find();
```

Node.js

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Node.js verwenden.

```
// Node.js callback API:

const bankDB = await mongoclient.db("bank");
var accountColl = await bankDB.createCollection("account");
```

```
var amountToTransfer = 500;

const session = mongoclient.startSession({causalConsistency: false});
await accountColl.drop();

await accountColl.insertOne({name: "Alice", balance: 1000}, { session });
await accountColl.insertOne({name: "Bob", balance: 1000}, { session });

const transactionOptions = {
    readConcern: { level: 'snapshot' },
    writeConcern: { w: 'majority' }
    };

// deduct $500 from Alice's account
var aliceBalance = await accountColl.findOne({name: "Alice"}, {session});
assert(aliceBalance.balance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
session.startTransaction(transactionOptions);
await accountColl.updateOne({name: "Alice"}, {$set: {balance: newAliceBalance}},
 {session });
await session.commitTransaction();
aliceBalance = await accountColl.findOne({name: "Alice"}, {session});
assert(newAliceBalance == aliceBalance.balance);

// add $500 to Bob's account
var bobBalance = await accountColl.findOne({name: "Bob"}, {session});
var newBobBalance = bobBalance.balance + amountToTransfer;
session.startTransaction(transactionOptions);
await accountColl.updateOne({name: "Bob"}, {$set: {balance: newBobBalance}},
 {session });
await session.commitTransaction();
bobBalance = await accountColl.findOne({name: "Bob"}, {session});
assert(newBobBalance == bobBalance.balance);
```

C#

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit C# verwenden.

```
// C# Callback API

var dbName = "bank";
var collName = "account";
var amountToTransfer = 500;
```

```
using (var session = client.StartSession(new ClientSessionOptions{CausalConsistency
 = false}))
{
    var bankDB = client.GetDatabase(dbName);
    var accountColl = bankDB.GetCollection<BsonDocument>(collName);
    bankDB.DropCollection(collName);
    accountColl.InsertOne(session, new BsonDocument { {"name", "Alice"}, {"balance",
 1000 } });
    accountColl.InsertOne(session, new BsonDocument { {"name", "Bob"}, {"balance",
 1000 } });

    // start transaction
    var transactionOptions = new TransactionOptions(
            readConcern: ReadConcern.Snapshot,
            writeConcern: WriteConcern.WMajority);
    var result = session.WithTransaction(
        (sess, cancellationtoken) =>
        {
            // deduct $500 from Alice's account
            var aliceBalance = accountColl.Find(sess,
 Builders<BsonDocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
            Debug.Assert(aliceBalance >= amountToTransfer);
            var newAliceBalance = aliceBalance.AsInt32 - amountToTransfer;
            accountColl.UpdateOne(sess, Builders<BsonDocument>.Filter.Eq("name",
 "Alice"),
                                  Builders<BsonDocument>.Update.Set("balance",
 newAliceBalance));
            aliceBalance = accountColl.Find(sess,
 Builders<BsonDocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
            Debug.Assert(aliceBalance == newAliceBalance);

            // add $500 from Bob's account
            var bobBalance = accountColl.Find(sess,
 Builders<BsonDocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
            var newBobBalance = bobBalance.AsInt32 + amountToTransfer;
            accountColl.UpdateOne(sess, Builders<BsonDocument>.Filter.Eq("name",
 "Bob"),
                                  Builders<BsonDocument>.Update.Set("balance",
 newBobBalance));
```

```
            bobBalance = accountColl.Find(sess,
 Builders<BsonDocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
            Debug.Assert(bobBalance == newBobBalance);

            return "Transaction committed";
        }, transactionOptions);
   // check values outside of transaction
    var aliceNewBalance = accountColl.Find(Builders<BsonDocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
    var bobNewBalance = accountColl.Find(Builders<BsonDocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
    Debug.Assert(aliceNewBalance ==  500);
    Debug.Assert(bobNewBalance ==  1500);
}
```

Ruby

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Ruby verwenden.

```
// Ruby Callback API

dbName = "bank"
collName = "account"
amountToTransfer = 500

session = client.start_session(:causal_consistency=> false)
bankDB = Mongo::Database.new(client, dbName)
accountColl = bankDB[collName]
accountColl.drop()

accountColl.insert_one({"name"=>"Alice", "balance"=>1000})
accountColl.insert_one({"name"=>"Bob", "balance"=>1000})

    # start transaction
    session.with_transaction(read_concern: {level: :snapshot}, write_concern:
 {w: :majority}) do
        # deduct $500 from Alice's account
        aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
 session).first['balance']
        assert aliceBalance >= amountToTransfer
        newAliceBalance = aliceBalance - amountToTransfer
```

```
        accountColl.update_one({"name"=>"Alice"}, { "$set" =>
 {"balance"=>newAliceBalance} }, :session=> session)
        aliceBalance = accountColl.find({"name"=>>"Alice"}, :session=>
 session).first['balance']
        assert_equal(newAliceBalance, aliceBalance)

        # add $500 from Bob's account
        bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
        newBobBalance = bobBalance + amountToTransfer
        accountColl.update_one({"name"=>"Bob"}, { "$set" =>
 {"balance"=>newBobBalance} }, :session=> session)
        bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
        assert_equal(newBobBalance, bobBalance)
    end

   # check results outside of transaction
    aliceBalance = accountColl.find({"name"=>"Alice"}).first['balance']
    bobBalance = accountColl.find({"name"=>"Bob"}).first['balance']
    assert_equal(aliceBalance, 500)
    assert_equal(bobBalance, 1500)

session.end_session
```

Go

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Go verwenden.

```
// Go - Callback API
type Account struct {
    Name string
    Balance  int
}

ctx := context.TODO()

dbName := "bank"
collName := "account"
amountToTransfer := 500

session, err := client.StartSession(options.Session().SetCausalConsistency(false))
assert.NilError(t, err)
defer session.EndSession(ctx)
```

```
bankDB := client.Database(dbName)
accountColl := bankDB.Collection(collName)
accountColl.Drop(ctx)

_, err = accountColl.InsertOne(ctx, bson.M{"name" : "Alice", "balance":1000})
_, err = accountColl.InsertOne(ctx, bson.M{"name" : "Bob", "balance":1000})

transactionOptions := options.Transaction().SetReadConcern(readconcern.Snapshot()).

 SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
_, err = session.WithTransaction(ctx, func(sessionCtx mongo.SessionContext)
 (interface{}, error) {
    var result Account
    // deduct $500 from Alice's account
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
    _, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Alice"},
 bson.M{"$set": bson.M{"balance": newAliceBalance}})
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance = result.Balance
    assert.Equal(t, aliceBalance, newAliceBalance)

    // add $500 to Bob's account
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    newBobBalance := bobBalance + amountToTransfer
    _, err = accountColl.UpdateOne(sessionCtx, bson.M{"name": "Bob"}, bson.M{"$set":
 bson.M{"balance": newBobBalance}})
    err = accountColl.FindOne(sessionCtx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance = result.Balance
    assert.Equal(t, bobBalance, newBobBalance)

    if err != nil {
        return nil, err
    }
    return "transaction committed", err
}, transactionOptions)

// check results outside of transaction
var result Account
err = accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&result)
aliceNewBalance := result.Balance
```

```
err = accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
bobNewBalance := result.Balance
assert.Equal(t, aliceNewBalance, 500)
assert.Equal(t, bobNewBalance, 1500)
// Go - Core API
type Account struct {
    Name string
    Balance  int
}

func transferMoneyWithRetry(sessionContext mongo.SessionContext, accountColl
 *mongo.Collection, t *testing.T) error {
    amountToTransfer := 500

    transactionOptions :=
 options.Transaction().SetReadConcern(readconcern.Snapshot()).

 SetWriteConcern(writeconcern.New(writeconcern.WMajority()))
    if err := sessionContext.StartTransaction(transactionOptions); err != nil {
        panic(err)
    }

    var result Account
    // deduct $500 from Alice's account
    err := accountColl.FindOne(sessionContext, bson.M{"name":
 "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Alice"},
 bson.M{"$set": bson.M{"balance": newAliceBalance}})
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name":
 "Alice"}).Decode(&result)
    aliceBalance = result.Balance
    assert.Equal(t, aliceBalance, newAliceBalance)

    // add $500 to Bob's account
    err = accountColl.FindOne(sessionContext, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    newBobBalance := bobBalance + amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Bob"},
 bson.M{"$set": bson.M{"balance": newBobBalance}})
```

```
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance = result.Balance
    assert.Equal(t, bobBalance, newBobBalance)

    err = sessionContext.CommitTransaction(sessionContext)
    return err
}

func doTransactionWithRetry(t *testing.T) {
    ctx := context.TODO()

    dbName := "bank"
    collName := "account"
    bankDB := client.Database(dbName)
    accountColl := bankDB.Collection(collName)

    client.UseSessionWithOptions(ctx, options.Session().SetCausalConsistency(false),
 func(sessionContext mongo.SessionContext) error {
        accountColl.Drop(ctx)
        accountColl.InsertOne(sessionContext, bson.M{"name" : "Alice",
 "balance":1000})
        accountColl.InsertOne(sessionContext, bson.M{"name" : "Bob",
 "balance":1000})
        for {
            err := transferMoneyWithRetry(sessionContext, accountColl, t)
            if err == nil {
                println("transaction committed")
                return nil
            }
            if mongoErr := err.(mongo.CommandError);
 mongoErr.HasErrorLabel("TransientTransactionError") {
                continue
            }
            println("transaction failed")
            return err
        }
    })

    // check results outside of transaction
    var result Account
    accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&esult)
```

```
    aliceBalance := result.Balance
    assert.Equal(t, aliceBalance, 500)
    accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    assert.Equal(t, bobBalance, 1500)
}
```

Java

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Java verwenden.

```
// Java (sync) - Callback API
MongoDatabase bankDB = mongoClient.getDatabase("bank");
MongoCollection accountColl = bankDB.getCollection("account");
accountColl.drop();
int amountToTransfer = 500;

// add sample data
accountColl.insertOne(new Document("name", "Alice").append("balance", 1000));
accountColl.insertOne(new Document("name", "Bob").append("balance", 1000));

TransactionOptions txnOptions = TransactionOptions.builder()
        .readConcern(ReadConcern.SNAPSHOT)
        .writeConcern(WriteConcern.MAJORITY)
        .build();
ClientSessionOptions sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build();
try ( ClientSession clientSession = mongoClient.startSession(sessionOptions) ) {
    clientSession.withTransaction(new TransactionBody<Void>() {
        @Override
        public Void execute() {
            // deduct $500 from Alice's account
            List<Document> documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
            int aliceBalance = (int) documentList.get(0).get("balance");
            int newAliceBalance = aliceBalance - amountToTransfer;

            accountColl.updateOne(clientSession, new Document("name", "Alice"), new
 Document("$set", new Document("balance", newAliceBalance)));

            // check Alice's new balance
```

```
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
            int updatedBalance = (int) documentList.get(0).get("balance");
            Assert.assertEquals(updatedBalance, newAliceBalance);

            // add $500 to Bob's account
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
            int bobBalance = (int) documentList.get(0).get("balance");
            int newBobBalance = bobBalance + amountToTransfer;

            accountColl.updateOne(clientSession, new Document("name", "Bob"), new
 Document("$set", new Document("balance", newBobBalance)));

            // check Bob's new balance
            documentList = new ArrayList<>();
            accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
            updatedBalance = (int) documentList.get(0).get("balance");
            Assert.assertEquals(updatedBalance, newBobBalance);

            return null;
        }
    }, txnOptions);
}
```

C

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit C verwenden.

```
// Sample Code for C with Callback

#include <bson.h>
#include <mongoc.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>

typedef struct {
    int64_t balance;
    bson_t *account;
    bson_t *opts;
```

```
    mongoc_collection_t *collection;
} ctx_t;

bool callback_session (mongoc_client_session_t *session, void *ctx, bson_t **reply,
 bson_error_t *error)
{
    bool r = true;
    ctx_t *data = (ctx_t *) ctx;
    bson_t local_reply;
    bson_t *selector = data->account;
    bson_t *update = BCON_NEW ("$set", "{", "balance", BCON_INT64 (data->balance),
 "}");

    mongoc_collection_update_one (data->collection, selector, update, data->opts,
 &local_reply, error);

    *reply = bson_copy (&local_reply);
    bson_destroy (&local_reply);
    bson_destroy (update);
    return r;
}

void test_callback_money_transfer(mongoc_client_t* client, mongoc_collection_t*
 collection, int amount_to_transfer){

    bson_t reply;
    bool r = true;
    const bson_t *doc;
    bson_iter_t iter;
    ctx_t alice_ctx;
    ctx_t bob_ctx;
    bson_error_t error;

    // find query
    bson_t *alice_query = bson_new ();
    BSON_APPEND_UTF8(alice_query, "name", "Alice");

    bson_t *bob_query = bson_new ();
    BSON_APPEND_UTF8(bob_query, "name", "Bob");

    // create session
    // set causal consistency to false
    mongoc_session_opt_t *session_opts = mongoc_session_opts_new ();
    mongoc_session_opts_set_causal_consistency (session_opts, false);
```

```
   // start the session
   mongoc_client_session_t *client_session = mongoc_client_start_session (client,
session_opts, &error);

   // add session to options
   bson_t *opts = bson_new();
   mongoc_client_session_append (client_session, opts, &error);

   // deduct 500 from Alice
   // find account balance of Alice
   mongoc_cursor_t *cursor = mongoc_collection_find_with_opts (collection,
alice_query, NULL, NULL);
   mongoc_cursor_next (cursor, &doc);
   bson_iter_init (&iter, doc);
   bson_iter_find (&iter, "balance");
   int64_t alice_balance = (bson_iter_value (&iter))->value.v_int64;
   assert(alice_balance >= amount_to_transfer);
   int64_t new_alice_balance = alice_balance - amount_to_transfer;

   // set variables which will be used by callback function
   alice_ctx.collection = collection;
   alice_ctx.opts = opts;
   alice_ctx.balance = new_alice_balance;
   alice_ctx.account = alice_query;

   // callback
   r = mongoc_client_session_with_transaction (client_session, &callback_session,
NULL, &alice_ctx, &reply, &error);
   assert(r);

   // find account balance of Alice after transaction
   cursor = mongoc_collection_find_with_opts (collection, alice_query, NULL, NULL);
   mongoc_cursor_next (cursor, &doc);
   bson_iter_init (&iter, doc);
   bson_iter_find (&iter, "balance");
   alice_balance = (bson_iter_value (&iter))->value.v_int64;
   assert(alice_balance == new_alice_balance);
   assert(alice_balance == 500);

       // add 500 to bob's balance
   // find account balance of Bob
   cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
   mongoc_cursor_next (cursor, &doc);
   bson_iter_init (&iter, doc);
```

```
    bson_iter_find (&iter, "balance");
    int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
    int64_t new_bob_balance = bob_balance + amount_to_transfer;

    bob_ctx.collection = collection;
    bob_ctx.opts = opts;
    bob_ctx.balance = new_bob_balance;
    bob_ctx.account = bob_query;

    // set read & write concern
    mongoc_read_concern_t *read_concern = mongoc_read_concern_new ();
    mongoc_write_concern_t *write_concern = mongoc_write_concern_new ();
    mongoc_transaction_opt_t *txn_opts = mongoc_transaction_opts_new ();

    mongoc_write_concern_set_w(write_concern, MONGOC_WRITE_CONCERN_W_MAJORITY);
    mongoc_read_concern_set_level(read_concern, MONGOC_READ_CONCERN_LEVEL_SNAPSHOT);
    mongoc_transaction_opts_set_write_concern (txn_opts, write_concern);
    mongoc_transaction_opts_set_read_concern (txn_opts, read_concern);

    // callback
    r = mongoc_client_session_with_transaction (client_session, &callback_session,
 txn_opts, &bob_ctx, &reply, &error);
    assert(r);

 // find account balance of Bob after transaction
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    bob_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(bob_balance == new_bob_balance);
    assert(bob_balance == 1500);

    // cleanup
    bson_destroy(alice_query);
    bson_destroy(bob_query);
    mongoc_client_session_destroy(client_session);
    bson_destroy(opts);
    mongoc_transaction_opts_destroy(txn_opts);
    mongoc_read_concern_destroy(read_concern);
    mongoc_write_concern_destroy(write_concern);
    mongoc_cursor_destroy(cursor);
    bson_destroy(doc);
}
```

```
int main(int argc, char* argv[]) {
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
 "account");
    // set amount to transfer
    int64_t amount_to_transfer = 500;
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);

    // open Alice account
    bson_t *alice_account = bson_new ();
    BSON_APPEND_UTF8(alice_account, "name", "Alice");
    BSON_APPEND_INT64(alice_account, "balance", 1000);

    // open Bob account
    bson_t *bob_account = bson_new ();
    BSON_APPEND_UTF8(bob_account, "name", "Bob");
    BSON_APPEND_INT64(bob_account, "balance", 1000);

    bool r = true;

    r = mongoc_collection_insert_one(collection, alice_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}
    r = mongoc_collection_insert_one(collection, bob_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}

    test_callback_money_transfer(client, collection, amount_to_transfer);

}
```

Python

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Python verwenden.

```
// Sample Python code with callback api
```

```
import pymongo

def callback(session, balance, query):
    collection.update_one(query, {'$set': {"balance": balance}}, session=session)

client = pymongo.MongoClient(<connection uri>)
rc_snapshot = pymongo.read_concern.ReadConcern('snapshot')
wc_majority = pymongo.write_concern.WriteConcern('majority')

# To start, drop and create an account collection and insert balances for both Alice
 and Bob
collection = client.get_database("bank").get_collection("account")
collection.drop()
collection.insert_one({"_id": 1, "name": "Alice", "balance": 1000})
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500

# deduct 500 from Alice's account
alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert alice_balance >= amount_to_transfer
new_alice_balance = alice_balance - amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.with_transaction(lambda s: callback(s, new_alice_balance, {"name":
 "Alice"}), read_concern=rc_snapshot, write_concern=wc_majority)

updated_alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert updated_alice_balance == new_alice_balance

# add 500 to Bob's account
bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert bob_balance >= amount_to_transfer
new_bob_balance = bob_balance + amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.with_transaction(lambda s: callback(s, new_bob_balance, {"name":
 "Bob"}), read_concern=rc_snapshot, write_concern=wc_majority)

updated_bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert updated_bob_balance == new_bob_balance
Sample Python code with Core api
import pymongo
```

```
client = pymongo.MongoClient(<connection_string>)
rc_snapshot = pymongo.read_concern.ReadConcern('snapshot')
wc_majority = pymongo.write_concern.WriteConcern('majority')

# To start, drop and create an account collection and insert balances for both Alice
 and Bob
collection = client.get_database("bank").get_collection("account")
collection.drop()
collection.insert_one({"_id": 1, "name": "Alice", "balance": 1000})
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500

# deduct 500 from Alice's account
alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert alice_balance >= amount_to_transfer
new_alice_balance = alice_balance - amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.start_transaction(read_concern=rc_snapshot, write_concern=wc_majority)
    collection.update_one({"name": "Alice"}, {'$set': {"balance":
 new_alice_balance}}, session=session)
    session.commit_transaction()

updated_alice_balance = collection.find_one({"name": "Alice"}).get("balance")
assert updated_alice_balance == new_alice_balance

# add 500 to Bob's account
bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert bob_balance >= amount_to_transfer
new_bob_balance = bob_balance + amount_to_transfer

with client.start_session({'causalConsistency':False}) as session:
    session.start_transaction(read_concern=rc_snapshot, write_concern=wc_majority)
    collection.update_one({"name": "Bob"}, {'$set': {"balance": new_bob_balance}},
 session=session)
    session.commit_transaction()

updated_bob_balance = collection.find_one({"name": "Bob"}).get("balance")
assert updated_bob_balance == new_bob_balance
```

Beispiele für Transaktions-APIs für die Core-API

Javascript

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Javascript verwenden.

```
// *** Transfer $500 from Alice to Bob inside a transaction: Success ***
// Setup bank account for Alice and Bob. Each have $1000 in their account
var databaseName = "bank";
var collectionName = "account";
var amountToTransfer = 500;

var session = db.getMongo().startSession({causalConsistency: false});
var bankDB = session.getDatabase(databaseName);
var accountColl = bankDB[collectionName];
accountColl.drop();

accountColl.insert({name: "Alice", balance: 1000});
accountColl.insert({name: "Bob", balance: 1000});

session.startTransaction();

// deduct $500 from Alice's account
var aliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert(aliceBalance >= amountToTransfer);
var newAliceBalance = aliceBalance - amountToTransfer;
accountColl.update({"name": "Alice"},{"$set": {"balance": newAliceBalance}});
var findAliceBalance = accountColl.find({"name": "Alice"}).next().balance;
assert.eq(newAliceBalance, findAliceBalance);

// add $500 to Bob's account
var bobBalance = accountColl.find({"name": "Bob"}).next().balance;
var newBobBalance = bobBalance + amountToTransfer;
accountColl.update({"name": "Bob"},{"$set": {"balance": newBobBalance}});
var findBobBalance = accountColl.find({"name": "Bob"}).next().balance;
assert.eq(newBobBalance, findBobBalance);

session.commitTransaction();

accountColl.find();
```

C#

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit C# verwenden.

```
// C# Core API

public void TransferMoneyWithRetry(IMongoCollection<bSondocument> accountColl,
 IClientSessionHandle session)
{
    var amountToTransfer = 500;

    // start transaction
   var transactionOptions = new TransactionOptions(
                readConcern: ReadConcern.Snapshot,
                writeConcern: WriteConcern.WMajority);
    session.StartTransaction(transactionOptions);
   try
    {
        // deduct $500 from Alice's account
        var aliceBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
       Debug.Assert(aliceBalance >= amountToTransfer);
       var newAliceBalance = aliceBalance.AsInt32 - amountToTransfer;
       accountColl.UpdateOne(session, Builders<bSondocument>.Filter.Eq("name",
 "Alice"),
                                Builders<bSondocument>.Update.Set("balance",
 newAliceBalance));
       aliceBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
       Debug.Assert(aliceBalance == newAliceBalance);

       // add $500 from Bob's account
       var bobBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
       var newBobBalance = bobBalance.AsInt32 + amountToTransfer;
       accountColl.UpdateOne(session, Builders<bSondocument>.Filter.Eq("name",
 "Bob"),
                                Builders<bSondocument>.Update.Set("balance",
 newBobBalance));
```

```
        bobBalance = accountColl.Find(session,
 Builders<bSondocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
        Debug.Assert(bobBalance == newBobBalance);

    }
    catch (Exception e)
    {
        session.AbortTransaction();
        throw;
    }

    session.CommitTransaction();
 }

}
public void DoTransactionWithRetry(MongoClient client)
{
    var dbName = "bank";
    var collName = "account";
    using (var session = client.StartSession(new
 ClientSessionOptions{CausalConsistency = false}))
    {
        try
        {
            var bankDB = client.GetDatabase(dbName);
            var accountColl = bankDB.GetCollection<bSondocument>(collName);
            bankDB.DropCollection(collName);
            accountColl.InsertOne(session, new BsonDocument { {"name", "Alice"},
 {"balance", 1000 } });
            accountColl.InsertOne(session, new BsonDocument { {"name", "Bob"},
 {"balance", 1000 } });

            while(true) {
                try
                {
                        TransferMoneyWithRetry(accountColl, session);
                        break;
                }
                catch (MongoException e)
                {
                    if(e.HasErrorLabel("TransientTransactionError"))
                    {
                        continue;
```

```
                    }
                    else
                    {
                        throw;
                    }
                }
            }

            // check values outside of transaction
            var aliceNewBalance =
 accountColl.Find(Builders<bSondocument>.Filter.Eq("name",
 "Alice")).FirstOrDefault().GetValue("balance");
            var bobNewBalance =
 accountColl.Find(Builders<bSondocument>.Filter.Eq("name",
 "Bob")).FirstOrDefault().GetValue("balance");
            Debug.Assert(aliceNewBalance ==  500);
            Debug.Assert(bobNewBalance ==  1500);
        }
        catch (Exception e)
        {
            Console.WriteLine("Error running transaction: " + e.Message);
        }
    }
}
```

Ruby

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Ruby verwenden.

```
# Ruby Core API

def transfer_money_w_retry(session, accountColl)
    amountToTransfer = 500

    session.start_transaction(read_concern: {level: :snapshot}, write_concern:
 {w: :majority})
    # deduct $500 from Alice's account
    aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
 session).first['balance']
    assert aliceBalance >= amountToTransfer
    newAliceBalance = aliceBalance - amountToTransfer
```

```
    accountColl.update_one({"name"=>"Alice"}, { "$set" =>
 {"balance"=>newAliceBalance} }, :session=> session)
    aliceBalance = accountColl.find({"name"=>"Alice"}, :session=>
 session).first['balance']
    assert_equal(newAliceBalance, aliceBalance)

    # add $500 to Bob's account
    bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
    newBobBalance = bobBalance + amountToTransfer
    accountColl.update_one({"name"=>"Bob"}, { "$set" =>
 {"balance"=>newBobBalance} }, :session=> session)
    bobBalance = accountColl.find({"name"=>"Bob"}, :session=>
 session).first['balance']
    assert_equal(newBobBalance, bobBalance)

    session.commit_transaction

end

def do_txn_w_retry(client)
     dbName = "bank"
    collName = "account"

    session = client.start_session(:causal_consistency=> false)
    bankDB = Mongo::Database.new(client, dbName)
    accountColl = bankDB[collName]
    accountColl.drop()

    accountColl.insert_one({"name"=>"Alice", "balance"=>1000})
    accountColl.insert_one({"name"=>"Bob", "balance"=>1000})

    begin
        transferMoneyWithRetry(session, accountColl)
        puts "transaction committed"
    rescue Mongo::Error => e
        if e.label?('TransientTransactionError')
            retry
        else
            puts "transaction failed"
            raise
        end
    end
```

```
    # check results outside of transaction
    aliceBalance = accountColl.find({"name"=>"Alice"}).first['balance']
    bobBalance = accountColl.find({"name"=>"Bob"}).first['balance']
    assert_equal(aliceBalance, 500)
    assert_equal(bobBalance, 1500)

end
```

Java

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Java verwenden.

```
// Java (sync) - Core API

public void transferMoneyWithRetry() {
   // connect to server
    MongoClientURI mongoURI = new MongoClientURI(uri);
    MongoClient mongoClient = new MongoClient(mongoURI);

    MongoDatabase bankDB = mongoClient.getDatabase("bank");
    MongoCollection accountColl = bankDB.getCollection("account");
    accountColl.drop();

   // insert some sample data
    accountColl.insertOne(new Document("name", "Alice").append("balance", 1000));
    accountColl.insertOne(new Document("name", "Bob").append("balance", 1000));

    while (true) {
        try {
            doTransferMoneyWithRetry(accountColl, mongoClient);
            break;
        } catch (MongoException e) {
            if (e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL)) {
                continue;
            } else {
                throw e;
            }
        }
    }
}
```

```
public void doTransferMoneyWithRetry(MongoCollection accountColl, MongoClient
 mongoClient) {
    int amountToTransfer = 500;

   TransactionOptions txnOptions = TransactionOptions.builder()
      .readConcern(ReadConcern.SNAPSHOT)
      .writeConcern(WriteConcern.MAJORITY)
      .build();
    ClientSessionOptions sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build();
    try ( ClientSession clientSession = mongoClient.startSession(sessionOptions) ) {
        clientSession.startTransaction(txnOptions);

        // deduct $500 from Alice's account
        List<Document> documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
        int aliceBalance = (int) documentList.get(0).get("balance");
        Assert.assertTrue(aliceBalance >= amountToTransfer);
        int newAliceBalance = aliceBalance - amountToTransfer;
        accountColl.updateOne(clientSession, new Document("name", "Alice"), new
 Document("$set", new Document("balance", newAliceBalance)));

        // check Alice's new balance
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Alice")).into(documentList);
        int updatedBalance = (int) documentList.get(0).get("balance");
        Assert.assertEquals(updatedBalance, newAliceBalance);

        // add $500 to Bob's account
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
        int bobBalance = (int) documentList.get(0).get("balance");
        int newBobBalance = bobBalance + amountToTransfer;
        accountColl.updateOne(clientSession, new Document("name", "Bob"), new
 Document("$set", new Document("balance", newBobBalance)));

        // check Bob's new balance
        documentList = new ArrayList<>();
        accountColl.find(clientSession, new Document("name",
 "Bob")).into(documentList);
        updatedBalance = (int) documentList.get(0).get("balance");
```

```
        Assert.assertEquals(updatedBalance, newBobBalance);

        // commit transaction
        clientSession.commitTransaction();
    }
}
// Java (async) -- Core API
public void transferMoneyWithRetry() {
    // connect to the server
    MongoClient mongoClient = MongoClients.create(uri);

    MongoDatabase bankDB = mongoClient.getDatabase("bank");
    MongoCollection accountColl = bankDB.getCollection("account");
    SubscriberLatchWrapper<Void> dropCallback = new SubscriberLatchWrapper<>();
    mongoClient.getDatabase("bank").drop().subscribe(dropCallback);
    dropCallback.await();

    // insert some sample data
    SubscriberLatchWrapper<InsertOneResult> insertionCallback = new
 SubscriberLatchWrapper<>();
    accountColl.insertOne(new Document("name", "Alice").append("balance",
 1000)).subscribe(insertionCallback);
    insertionCallback.await();

    insertionCallback = new SubscriberLatchWrapper<>();
    accountColl.insertOne(new Document("name", "Bob").append("balance",
 1000)).subscribe(insertionCallback);;
    insertionCallback.await();

    while (true) {
        try {
            doTransferMoneyWithRetry(accountColl, mongoClient);
            break;
        } catch (MongoException e) {
            if (e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL)) {
                continue;
            } else {
                throw e;
            }
        }
    }
}
```

```
public void doTransferMoneyWithRetry(MongoCollection accountColl, MongoClient
 mongoClient) {
    int amountToTransfer = 500;

    // start the transaction
    TransactionOptions txnOptions = TransactionOptions.builder()
            .readConcern(ReadConcern.SNAPSHOT)
            .writeConcern(WriteConcern.MAJORITY)
            .build();
    ClientSessionOptions sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build();

    SubscriberLatchWrapper<ClientSession> sessionCallback = new
 SubscriberLatchWrapper<>();
    mongoClient.startSession(sessionOptions).subscribe(sessionCallback);
    ClientSession session = sessionCallback.get().get(0);
    session.startTransaction(txnOptions);

    // deduct $500 from Alice's account
    SubscriberLatchWrapper<Document> findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Alice")).first().subscribe(findCallback);
    Document documentFound = findCallback.get().get(0);
    int aliceBalance = (int) documentFound.get("balance");
    int newAliceBalance = aliceBalance - amountToTransfer;

    SubscriberLatchWrapper<UpdateResult> updateCallback = new
 SubscriberLatchWrapper<>();
    accountColl.updateOne(session, new Document("name",
 "Alice"), new Document("$set", new Document("balance",
 newAliceBalance))).subscribe(updateCallback);
    updateCallback.await();

    // check Alice's new balance
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Alice")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int updatedBalance = (int) documentFound.get("balance");
    Assert.assertEquals(updatedBalance, newAliceBalance);

    // add $500 to Bob's account
    findCallback = new SubscriberLatchWrapper<>();
```

```
    accountColl.find(session, new Document("name",
 "Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int bobBalance = (int) documentFound.get("balance");
    int newBobBalance = bobBalance + amountToTransfer;

    updateCallback = new SubscriberLatchWrapper<>();
    accountColl.updateOne(session, new Document("name", "Bob"), new Document("$set",
 new Document("balance", newBobBalance))).subscribe(updateCallback);
    updateCallback.await();

    // check Bob's new balance
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    updatedBalance = (int) documentFound.get("balance");
    Assert.assertEquals(updatedBalance, newBobBalance);

    // commit the transaction
    SubscriberLatchWrapper<Void> transactionCallback = new
 SubscriberLatchWrapper<>();
    session.commitTransaction().subscribe(transactionCallback);
    transactionCallback.await();
}

public class SubscriberLatchWrapper<T> implements Subscriber<T> {

    /**
     * A Subscriber that stores the publishers results and provides a latch so can
 block on completion.
     *
     * @param <T> The publishers result type
     */
    private final List<T> received;
    private final List<RuntimeException> errors;
    private final CountDownLatch latch;
    private volatile Subscription subscription;
    private volatile boolean completed;

    /**
     * Construct an instance
     */
    public SubscriberLatchWrapper() {
```

```
        this.received = new ArrayList<>();
        this.errors = new ArrayList<>();
        this.latch = new CountDownLatch(1);
    }

    @Override
    public void onSubscribe(final Subscription s) {
        subscription = s;
        subscription.request(Integer.MAX_VALUE);
    }

    @Override
    public void onNext(final T t) {
        received.add(t);
    }

    @Override
    public void onError(final Throwable t) {
        if (t instanceof RuntimeException) {
            errors.add((RuntimeException) t);
        } else {
            errors.add(new RuntimeException("Unexpected exception", t));
        }
        onComplete();
    }

    @Override
    public void onComplete() {
        completed = true;
        subscription.cancel();
        latch.countDown();
    }

    /**
     * Get received elements
     *
     * @return the list of received elements
     */
    public List<T> getReceived() {
        return received;
    }

    /**
     * Get received elements.
```

```
     *
     * @return the list of receive elements
     */
    public List<T> get() {
        return await().getReceived();
    }

    /**
     * Await completion or error
     *
     * @return this
     */
    public SubscriberLatchWrapper<T> await() {
        subscription.request(Integer.MAX_VALUE);
        try {
            if (!latch.await(300, TimeUnit.SECONDS)) {
                throw new MongoTimeoutException("Publisher onComplete timed out for
 300 seconds");
            }
        } catch (InterruptedException e) {
            throw new MongoInterruptedException("Interrupted waiting for
 observeration", e);
        }
        if (!errors.isEmpty()) {
            throw errors.get(0);
        }
        return this;
    }

    public boolean getCompleted() {
        return this.completed;
    }

    public void close() {
        subscription.cancel();
        received.clear();
    }
}
```

C

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit C verwenden.

```
// Sample C code with core session

bool core_session(mongoc_client_session_t *client_session, mongoc_collection_t*
 collection, bson_t *selector, int64_t balance){
    bool r = true;
    bson_error_t error;
    bson_t *opts = bson_new();
    bson_t *update = BCON_NEW ("$set", "{", "balance", BCON_INT64 (balance), "}");

    // set read & write concern
    mongoc_read_concern_t *read_concern = mongoc_read_concern_new ();
    mongoc_write_concern_t *write_concern = mongoc_write_concern_new ();
    mongoc_transaction_opt_t *txn_opts = mongoc_transaction_opts_new ();

    mongoc_write_concern_set_w(write_concern, MONGOC_WRITE_CONCERN_W_MAJORITY);
    mongoc_read_concern_set_level(read_concern, MONGOC_READ_CONCERN_LEVEL_SNAPSHOT);
    mongoc_transaction_opts_set_write_concern (txn_opts, write_concern);
    mongoc_transaction_opts_set_read_concern (txn_opts, read_concern);

    mongoc_client_session_start_transaction (client_session, txn_opts, &error);
    mongoc_client_session_append (client_session, opts, &error);

    r = mongoc_collection_update_one (collection, selector, update, opts, NULL,
 &error);

    mongoc_client_session_commit_transaction (client_session, NULL, &error);
    bson_destroy (opts);
    mongoc_transaction_opts_destroy(txn_opts);
    mongoc_read_concern_destroy(read_concern);
    mongoc_write_concern_destroy(write_concern);
    bson_destroy (update);
    return r;
}

void test_core_money_transfer(mongoc_client_t* client, mongoc_collection_t*
 collection, int amount_to_transfer){

    bson_t reply;
    bool r = true;
    const bson_t *doc;
    bson_iter_t iter;
    bson_error_t error;
```

```
    // find query
    bson_t *alice_query = bson_new ();
    BSON_APPEND_UTF8(alice_query, "name", "Alice");

    bson_t *bob_query = bson_new ();
    BSON_APPEND_UTF8(bob_query, "name", "Bob");

    // create session
    // set causal consistency to false
    mongoc_session_opt_t *session_opts = mongoc_session_opts_new ();
    mongoc_session_opts_set_causal_consistency (session_opts, false);
    // start the session
    mongoc_client_session_t *client_session = mongoc_client_start_session (client,
session_opts, &error);

    // add session to options
    bson_t *opts = bson_new();
    mongoc_client_session_append (client_session, opts, &error);

    // deduct 500 from Alice
    // find account balance of Alice
    mongoc_cursor_t *cursor = mongoc_collection_find_with_opts (collection,
alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance >= amount_to_transfer);
    int64_t new_alice_balance = alice_balance - amount_to_transfer;

    // core
    r = core_session (client_session, collection, alice_query, new_alice_balance);
    assert(r);

    // find account balance of Alice after transaction
    cursor = mongoc_collection_find_with_opts (collection, alice_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    alice_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(alice_balance == new_alice_balance);
    assert(alice_balance == 500);

    // add 500 to Bob's balance
```

```
    // find account balance of Bob
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
    int64_t new_bob_balance = bob_balance + amount_to_transfer;

    //core
    r = core_session (client_session, collection, bob_query, new_bob_balance);
    assert(r);

    // find account balance of Bob after transaction
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    bob_balance = (bson_iter_value (&iter))->value.v_int64;
    assert(bob_balance == new_bob_balance);
    assert(bob_balance == 1500);

    // cleanup
    bson_destroy(alice_query);
    bson_destroy(bob_query);
    mongoc_client_session_destroy(client_session);
    bson_destroy(opts);
    mongoc_cursor_destroy(cursor);
    bson_destroy(doc);
}

int main(int argc, char* argv[]) {
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
 "account");
    // set amount to transfer
    int64_t amount_to_transfer = 500;
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);
```

```
    // open Alice account
    bson_t *alice_account = bson_new ();
    BSON_APPEND_UTF8(alice_account, "name", "Alice");
    BSON_APPEND_INT64(alice_account, "balance", 1000);

    // open Bob account
    bson_t *bob_account = bson_new ();
    BSON_APPEND_UTF8(bob_account, "name", "Bob");
    BSON_APPEND_INT64(bob_account, "balance", 1000);

    bool r = true;

    r = mongoc_collection_insert_one(collection, alice_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}
    r = mongoc_collection_insert_one(collection, bob_account, NULL, NULL, &error);
    if (!r) {printf("Error encountered:%s", error.message);}

    test_core_money_transfer(client, collection, amount_to_transfer);

}
```

Scala

Der folgende Code zeigt, wie Sie die Amazon DocumentDB-Transaktions-API mit Scala verwenden.

```
// Scala Core API
def transferMoneyWithRetry(sessionObservable: SingleObservable[ClientSession] ,
 database: MongoDatabase ): Unit = {
    val accountColl = database.getCollection("account")
    var amountToTransfer = 500

    var transactionObservable: Observable[ClientSession] =
 sessionObservable.map(clientSession => {
    clientSession.startTransaction()

    // deduct $500 from Alice's account
    var aliceBalance = accountColl.find(clientSession, Document("name" ->
 "Alice")).await().head.getInteger("balance")
    assert(aliceBalance >= amountToTransfer)
    var newAliceBalance = aliceBalance - amountToTransfer
```

```
    accountColl.updateOne(clientSession, Document("name" -> "Alice"),
 Document("$set" -> Document("balance" -> newAliceBalance))).await()
    aliceBalance = accountColl.find(clientSession, Document("name" ->
 "Alice")).await().head.getInteger("balance")
    assert(aliceBalance == newAliceBalance)

    // add $500 to Bob's account
    var bobBalance = accountColl.find(clientSession, Document("name" ->
 "Bob")).await().head.getInteger("balance")
    var newBobBalance = bobBalance + amountToTransfer
    accountColl.updateOne(clientSession, Document("name" -> "Bob"), Document("$set"
 -> Document("balance" -> newBobBalance))).await()
    bobBalance = accountColl.find(clientSession, Document("name" ->
 "Bob")).await().head.getInteger("balance")
    assert(bobBalance == newBobBalance)

    clientSession
    })

    transactionObservable.flatMap(clientSession =>
 clientSession.commitTransaction()).await()
}

def doTransactionWithRetry(): Unit = {
    val client: MongoClient = MongoClientWrapper.getMongoClient()
    val database: MongoDatabase = client.getDatabase("bank")
    val accountColl = database.getCollection("account")
    accountColl.drop().await()

    val sessionOptions =
 ClientSessionOptions.builder().causallyConsistent(false).build()
    var  sessionObservable: SingleObservable[ClientSession] =
 client.startSession(sessionOptions)
    accountColl.insertOne(Document("name" -> "Alice", "balance" -> 1000)).await()
    accountColl.insertOne(Document("name" -> "Bob", "balance" -> 1000)).await()

    var retry = true
    while (retry) {
        try {
        transferMoneyWithRetry(sessionObservable, database)
        println("transaction committed")
        retry = false
        }
        catch {
```

```
        case e: MongoException if
 e.hasErrorLabel(MongoException.TRANSIENT_TRANSACTION_ERROR_LABEL) => {
            println("retrying transaction")
        }
        case other: Throwable => {
            println("transaction failed")
            retry = false
            throw other

        }
        }
    }

    // check results outside of transaction
    assert(accountColl.find(Document("name" ->
 "Alice")).results().head.getInteger("balance") == 500)
    assert(accountColl.find(Document("name" ->
 "Bob")).results().head.getInteger("balance") == 1500)

    accountColl.drop().await()

}
```

Unterstützte -Befehle

Befehl	Unterstützt
abortTransaction	Ja
commitTransaction	Ja
endSessions	Ja
killSession	Ja
killAllSession	Ja
killAllSessionsByPattern	Nein
refreshSessions	Nein

Befehl	Unterstützt
`startSession`	Ja

Nicht unterstützte Funktionen

Methoden	Stufen oder Befehle

`db.collection.aggregate()`	`$collStats`
	`$currentOp`
	`$indexStats`
	`$listSessions`
	`$out`

`db.collection.count()`
`db.collection.countDocuments()`
	`$where`
	`$near`
	`$nearSphere`

`db.collection.insert()`	`insert`wird nicht unterstützt, wenn es nicht für eine vorhandene Sammlung ausgeführt wird. Diese Methode wird unterstützt, wenn sie auf eine bereits vorhandene Sammlung abzielt.

Sitzungen

MongoDB-Sitzungen sind ein Framework, das zur Unterstützung wiederholbarer Schreibvorgänge, kausaler Konsistenz, Transaktionen und zur datenbankübergreifenden Verwaltung von Vorgängen verwendet wird. Wenn eine Sitzung erstellt wird, wird vom Client ein logischer Sitzungsbezeichner (lsid) generiert, der verwendet wird, um alle Vorgänge innerhalb dieser Sitzung zu taggen, wenn Befehle an den Server gesendet werden.

Amazon DocumentDB unterstützt die Verwendung von Sitzungen, um Transaktionen zu ermöglichen, unterstützt jedoch keine kausale Konsistenz oder wiederholbare Schreibvorgänge.

Wenn Sie Transaktionen in Amazon DocumentDB verwenden, wird eine Transaktion innerhalb einer Sitzung mithilfe der`session.startTransaction()` API initiiert, und eine Sitzung unterstützt jeweils eine einzelne Transaktion. In ähnlicher Weise werden Transaktionen entweder mit den APIs commit (`session.commitTransaction()`) oder abort (`session.abortTransaction()`) abgeschlossen.

Kausale Konsistente

Die kausale Konsistenz garantiert, dass der Client innerhalb einer einzelnen Clientsitzung die read-after-write Konsistenz einhält, monatomische Lese- und Schreibvorgänge auf Lesevorgänge folgen, und diese Garantien gelten für alle Instanzen in einem Cluster, nicht nur für die primären. Amazon DocumentDB unterstützt keine kausale Konsistenz und die folgende Aussage führt zu einem Fehler.

```
var mySession = db.getMongo().startSession();
var mySessionObject = mySession.getDatabase('test').getCollection('account');

mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});
//Result:{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

mySessionObject.find()
//Error: error: {
//         "ok" : 0,
//         "code" : 303,
//         "errmsg" : "Feature not supported: 'causal consistency'",
//         "operationTime" : Timestamp(1603461817, 493214)
//}

mySession.endSession()
```

Sie können die kausale Konsistenz innerhalb einer Sitzung deaktivieren. Bitte beachten Sie, dass Sie auf diese Weise das Sitzungsframework nutzen können, jedoch keine kausalen Konsistenzgarantien für Lesevorgänge bieten. Wenn Sie Amazon DocumentDB verwenden, sind Lesevorgänge von der primären Instanz read-after-write konsistent und Lesevorgänge aus den Replikat-Instances letztendlich konsistent. Transaktionen sind der primäre Anwendungsfall für die Nutzung von Sitzungen.

```
var mySession = db.getMongo().startSession({causalConsistency: false});
var mySessionObject = mySession.getDatabase('test').getCollection('account');

mySessionObject.updateOne({"_id": 2}, {"$inc": {"balance": 400}});
//Result:{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

mySessionObject.find()
//{ "_id" : 1, "name" : "Bob", "balance" : 100 }
//{ "_id" : 2, "name" : "Alice", "balance" : 1700 }
```

Wiederholbare Schreibvorgänge

Wiederholbares Schreiben ist eine Funktion, bei der der Client versucht, Schreibvorgänge einmal zu wiederholen, wenn Netzwerkfehler auftreten oder der Client den primären Schreibvorgang nicht finden kann. In Amazon DocumentDB werden wiederholbare Schreibvorgänge nicht unterstützt und müssen deaktiviert werden. Sie können es mit dem Befehl (`retryWrites=false`) in der Verbindungszeichenfolge deaktivieren.

Ausnahme: Wenn Sie Mongo Shell verwenden, fügen Sie den`retryWrites=false` Befehl in keine Codezeichenfolge ein. Standardmäßig werden wiederholbare Schreibvorgänge deaktiviert. Das`retryWrites=false` Einbeziehen kann zu Fehlern bei normalen Lesebefehlen führen.

Transaktionsfehler

Bei der Verwendung von Transaktionen gibt es Szenarien, die zu einem Fehler führen können, der besagt, dass eine Transaktionsnummer mit keiner laufenden Transaktion übereinstimmt.

Der Fehler kann in mindestens zwei verschiedenen Szenarien generiert werden:

- After the one-minute transaction timeout.
- After an instance restart (due to patching, crash recovery, etc.), it is possible to receive this error even in cases where the transaction successfully committed. During an instance restart, the database can't tell the difference between a transaction that successfully completed versus a transaction that aborted. In other words, the transaction completion state is ambiguous.

Der beste Weg, um mit diesem Fehler umzugehen, besteht darin, transaktionale Updates idempotent zu machen, indem Sie beispielsweise den`$set` Mutator anstelle einer Inkrement-/Dekrementierungsoperation verwenden. Siehe unten:

```
{ "ok" : 0,
"operationTime" : Timestamp(1603938167, 1),
"code" : 251,
"errmsg" : "Given transaction number 1 does not match any in-progress transactions."
}
```

Bewährte Methoden für Amazon DocumentDB

Lernen Sie bewährte Methoden für die Arbeit mit Amazon DocumentDB kennen (mit MongoDB-Kompatibilität). Dieser Abschnitt wird fortlaufend aktualisiert, wenn neue bewährte Methoden identifiziert werden.

Themen

Grundlegende Anleitungen für die Ausführung

Im Folgenden finden Sie grundlegende Anleitungen, die bei der Arbeit mit Amazon DocumentDB befolgt werden sollten. Das Amazon DocumentDB Service Level Agreement erfordert, dass Sie diese Richtlinien befolgen.

- Stellen Sie einen Cluster bereit, der aus zwei oder mehr Amazon DocumentDB-Instances in zwei AWS Availability Zones besteht. Für Produktions-Workloads empfehlen wir die Bereitstellung eines Clusters, der aus drei oder mehr Amazon DocumentDB-Instances in drei Availability Zones besteht.
- Verwenden Sie die Services innerhalb der angegebenen Service Limits. Weitere Informationen finden Sie unter Kontingente und Limits für Amazon DocumentDB .
- Überwachen Sie Speicher, CPU, Verbindungen und Speichernutzung. Um Sie bei der Aufrechterhaltung der Systemleistung und -verfügbarkeit zu unterstützen, richten Sie Amazon so

ein, CloudWatch dass Sie benachrichtigt werden, wenn sich Nutzungsmuster ändern oder wenn Sie sich der Kapazität Ihrer Bereitstellung nähern.

- Skalieren Sie Ihre Instances, wenn Sie die Grenzen der Speicherkapazität beinahe erreicht haben. Ihre Instances sollten mit genügend Rechenressourcen (d. h. RAM, CPU) ausgestattet sein, um Nachfragesteigerungen seitens Ihrer Anwendungen bewältigen zu können.
- Legen Sie die Aufbewahrungszeitraum für Backups so fest, dass sie zu Ihrem Wiederherstellungsziel passt.
- Testen Sie den Failover für Ihren Cluster, um zu verstehen, wie lange der Vorgang für Ihren Anwendungsfall dauert. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Failover.
- Stellen Sie mit dem Cluster-Endpunkt (siehe Amazon DocumentDB-Endpunkte) und im Replikat-Set-Modus (siehe Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz) eine Verbindung zu Ihrem Amazon DocumentDB-Cluster her, um die Auswirkungen eines Failovers auf Ihre Anwendung zu minimieren.
- Wählen Sie eine Treibereinstellung für die Leseeinstellung aus, die die Leseskalierung maximiert und gleichzeitig die Anforderungen für die Lesekonsistenz Ihrer Anwendung erfüllt. Die Leseeinstellung `secondaryPreferred` ermöglicht Replica-Lesevorgänge, sodass die primäre Instance produktiver sein kann. Weitere Informationen finden Sie unter Leseeinstellungsoptionen.
- Entwerfen Sie Ihre Anwendung so, dass Sie im Falle von Netzwerk- und Datenbankfehlern stabil ist. Unterscheiden Sie mithilfe des Fehlermechanismus des Treibers zwischen vorübergehenden Fehlern und persistenten Fehlern. Wiederholen Sie den Vorgang bei vorübergehenden Fehlern mit einem exponentiellen Backoff-Mechanismus (bei Bedarf). Stellen Sie sicher, dass Ihre Anwendung bei der Implementierung von Logik für Wiederholversuche die Datenkonsistenz berücksichtigt.
- Aktivieren Sie den Cluster-Löschschutz für alle Produktions-Cluster oder Cluster mit wertvollen Daten. Bevor Sie einen Amazon DocumentDB-Cluster löschen, erstellen Sie einen endgültigen Snapshot. Wenn Sie Ressourcen mit bereitstellen AWS CloudFormation, aktivieren Sie den Beendigungsschutz. Weitere Informationen finden Sie unter Beendigungsschutz und Löschschutz.
- Beim Erstellen eines Amazon DocumentDB-Clusters ist die --engine-version ein optionaler Parameter, der standardmäßig die neueste Engine-Hauptversion verwendet. Die aktuelle Engine-Hauptversion ist 4.0.0. Wenn neue Engine-Hauptversionen veröffentlicht werden, wird die Standard-Engine-Version für die Engine-Version aktualisiert, um die letzte Engine-Hauptversion widerzuspiegeln. Daher empfehlen wir, für Produktions-Workloads und insbesondere für Workloads, die von Skripterstellung, Automatisierung oder AWS CloudFormation Vorlagen abhängig sind, die --Engine-Version explizit auf die gewünschte Hauptversion festzulegen.

Instance-Größenbestimmung

Einer der wichtigsten Aspekte bei der Auswahl einer Instance-Größe in Amazon DocumentDB ist die Menge an RAM für Ihren Cache. Amazon DocumentDB reserviert ein Drittel des RAM für eigene Services, was bedeutet, dass nur zwei Drittel des Instance-RAM für den Cache verfügbar sind. Daher ist es eine bewährte Methode für Amazon DocumentDB, einen Instance-Typ mit genügend RAM auszuwählen, um Ihren Arbeitssatz (d. h. Daten und Indizes) im Speicher zu integrieren. Durch die richtige Größe von Instances wird die Gesamtleistung optimiert und die E/A-Kosten möglicherweise minimiert. Sie können den Größenrechner von Amazon DocumentDB von Drittanbietern verwenden, um die Instance-Größe für einen bestimmten Workload zu schätzen.

Um festzustellen, ob der Arbeitssatz Ihrer Anwendung in den Arbeitsspeicher passt, überwachen Sie die `BufferCacheHitRatio` mit Amazon CloudWatch für jede Instance in einem Cluster, der unter Last steht.

Die `BufferCacheHitRatio` CloudWatch Metrik misst den Prozentsatz der Daten und Indizes, die aus dem Speichercache einer Instance bereitgestellt werden (im Vergleich zum Speichervolumen). Im Allgemeinen sollte der Wert von `BufferCacheHitRatio` so hoch wie möglich sein, da das Lesen von Daten aus dem Arbeitssatz-Speicher schneller und kostengünstiger ist als das Lesen vom Speicher-Volume. Obwohl es wünschenswert ist, `BufferCacheHitRatio` möglichst nahe bei 100% zu halten, hängt der beste erreichbare Wert von den Zugriffsmustern und den Leistungsanforderungen Ihrer Anwendung ab. Um das höchstmögliche `BufferCacheHitRatio` beizubehalten, wird empfohlen, dass die Instances in Ihrem Cluster über ausreichend RAM verfügen, damit Ihre Indizes und Arbeitsdatensätze in den Speicher passen.

Wenn Ihre Indizes nicht in den Speicher passen, wird Ihnen ein niedrigeres `BufferCacheHitRatio` angezeigt. Beim durchgängigen Lesen von der Festplatte fallen zusätzliche E/A-Kosten an, darüber hinaus ist es nicht so leistungsstark wie das Lesen aus dem Speicher. Wenn Ihr `BufferCacheHitRatio`-Verhältnis niedriger als erwartet ist, skalieren Sie die Instance-Größe für Ihren Cluster, um mehr RAM zur Verfügung zu stellen, damit die Arbeitssatzdaten in den Speicher passen. Wenn das Skalieren der Instance-Klasse zu einem enormen Anstieg im `BufferCacheHitRatio` führt, passte der Arbeitssatz Ihrer Anwendung nicht in den Speicher. Skalieren Sie weiter, bis sich `BufferCacheHitRatio` nach einer Skalierungsoperation nicht mehr drastisch erhöht. Weitere Informationen zur Überwachung von Instance-Metriken finden Sie unter Amazon DocumentDB-Metriken.

Abhängig von Ihren Workload- und Latenzanforderungen kann es akzeptabel sein, dass Ihre Anwendung während der stabilen Zustandsnutzung höhere `BufferCacheHitRatio`-Werte

aufweist, das `BufferCacheHitRatio` jedoch periodisch nachlässt, da analytische Abfragen, die eine gesamte Sammlung scannen müssen, auf einer Instance ausgeführt werden. Diese periodischen Einbrüche im `BufferCacheHitRatio` können sich als höhere Latenz für nachfolgende Abfragen manifestieren, die die Arbeitssatzdaten aus dem Speicher-Volume wieder in den Puffercache füllen müssen. Es wird empfohlen, Ihre Workloads zunächst in einer Vorproduktionsumgebung mit einem repräsentativen Produktions-Workload zu testen, um die Leistungsmerkmale und das **`BufferCacheHitRatio`** zu verstehen, bevor Sie den Workload für die Produktion bereitstellen.

Es handelt sich beim `BufferCacheHitRatio` um eine Instance-spezifische Metrik, daher können verschiedene Instances innerhalb desselben Clusters unterschiedliche `BufferCacheHitRatio`-Werte aufweisen, je nachdem, wie Lesevorgänge auf die Primär- und Replikat-Instances verteilt werden. Wenn Ihr betrieblicher Workload nicht mit periodischen Erhöhungen der Latenz durch erneutes Auffüllen des Arbeitssatzcache nach dem Ausführen analytischer Abfragen umgehen kann, sollten Sie versuchen, den Puffercache des regulären Workloads von dem der analytischen Abfragen zu isolieren. Sie können eine vollständige `BufferCacheHitRatio`-Isolation erreichen, indem Sie betriebliche Abfragen an die Primär-Instance und analytische Abfragen nur an die Replikat-Instances weiterleiten. Sie können auch eine partielle Isolation erreichen, indem Sie analytische Abfragen an eine bestimmte Replikat-Instance weiterleiten, mit dem Verständnis, dass ein gewisser Prozentsatz der regulären Abfragen auch auf diesem Replikat ausgeführt wird und möglicherweise betroffen sein könnte.

Angemessene `BufferCacheHitRatio`-Werte hängen von Ihrem Anwendungsfall und Ihren Anwendungsanforderungen ab. Es gibt keinen besten oder minimalen Wert für diese Metrik. Nur Sie können entscheiden, ob der Kompromiss von einem vorübergehend niedrigeren `BufferCacheHitRatio` aus Kosten- und Leistungsperspektive akzeptabel ist.

Arbeiten mit Indizes

Erstellen von Indizes

Beim Importieren von Daten in Amazon DocumentDB sollten Sie Ihre Indizes erstellen, bevor Sie große Datensätze importieren. Sie können das Amazon DocumentDB Index Tool verwenden, um Indizes aus einer laufenden MongoDB-Instance oder einem `mongodump` laufenden Verzeichnis zu extrahieren und diese Indizes in einem Amazon DocumentDB-Cluster zu erstellen. Weitere Hinweise zu Migrationen finden Sie unter Migration zu Amazon DocumentDB.

Index-Selektivität

Wir empfehlen Ihnen, die Erstellung von Indizes auf Felder zu beschränken, bei denen die Anzahl der Duplikatwerte weniger als 1 % der Gesamtzahl der Dokumente in der Sammlung beträgt. Wenn Ihre Sammlung beispielsweise 100.000 Dokumente enthält, erstellen Sie Indizes nur für Felder, in denen derselbe Wert 1.000 Mal oder weniger vorkommt.

Die Wahl eines Index mit einer hohen Anzahl von eindeutigen Werten (d. h. einer hohen Kardinalität) gewährleistet, dass Filteroperationen eine geringe Anzahl von Dokumenten zurückgeben, wodurch eine gute Leistung während der Index-Scans erzielt wird. Ein Beispiel für einen Index hoher Kardinalität ist ein einzigartiger Index, der garantiert, dass Gleichheitsprädikate höchstens ein einziges Dokument zurückgeben. Beispiele für niedrige Kardinalität sind ein Index über ein boolesches Feld und ein Index über einen Wochentag. Aufgrund ihrer schlechten Leistung ist es unwahrscheinlich, dass der Abfragenoptimierer der Datenbank Indizes mit niedriger Kardinalität auswählt. Gleichzeitig verbrauchen Indizes mit niedriger Kardinalität weiterhin Ressourcen wie Plattenplatz und E/A-Vorgänge. Als Faustregel gilt, dass Sie Indizes für Felder verwenden sollten, bei denen die typische Wertehäufigkeit 1 % der Gesamtsammlungsgröße oder weniger beträgt.

Darüber hinaus wird empfohlen, nur Indizes für Felder zu erstellen, die häufig als Filter verwendet werden, und regelmäßig nach nicht verwendeten Indizes zu suchen. Weitere Informationen finden Sie unter Wie analysiere ich die Indexnutzung und identifiziere ungenutzte Indizes?.

Auswirkungen von Indizes auf das Schreiben von Daten

Zwar können Indizes die Abfrageleistung verbessern, da nicht jedes Dokument in einer Sammlung gescannt werden muss, diese Verbesserung erfordert jedoch einen Kompromiss. Für jeden Index einer Sammlung muss die Datenbank jedes Mal, wenn ein Dokument eingefügt, aktualisiert oder gelöscht wird, die Sammlung aktualisieren und die Felder in jeden der Indizes für die Sammlung schreiben. Wenn eine Sammlung beispielsweise über neun Indizes verfügt, muss die Datenbank zehn Schreibvorgänge durchführen, bevor die Operation dem Kunden bestätigt wird. Daher verursacht jeder zusätzliche Index zusätzliche Schreiblatenz, E/As und eine Erhöhung des insgesamt genutzten Speichers.

Cluster-Instances müssen über eine angemessene Größe verfügen, um den gesamten Arbeitsspeicher zu erhalten. Dadurch wird vermieden, dass ständig Indexseiten aus dem Speichervolumen gelesen werden müssen, was sich negativ auf die Leistung auswirkt und höhere E/A-Kosten verursacht. Weitere Informationen finden Sie unter Instance-Größenbestimmung.

Um eine optimale Leistung zu erzielen, sollten Sie die Anzahl der Indizes in Ihren Sammlungen minimieren und nur die Indizes hinzufügen, die zur Verbesserung der Leistung bei häufigen Abfragen erforderlich sind. Bei variierenden Workloads gilt als gute Richtschnur, die Anzahl der Indizes pro Sammlung auf fünf oder weniger zu beschränken.

Identifizieren von fehlenden Indizes

Das Identifizieren fehlender Indizes ist eine bewährte Methode, die wir regelmäßig empfehlen. Weitere Informationen finden Sie unter Wie erkenne ich fehlende Indizes?.

Identifizieren von ungenutzten Indizes

Da das Identifizieren und Löschen fehlender Indizes eine bewährte Vorgehensweise ist, empfiehlt sich eine regelmäßige Durchführung. Weitere Informationen finden Sie unter Wie analysiere ich die Indexnutzung und identifiziere ungenutzte Indizes?.

Bewährte Methoden für die Sicherheit

Für bewährte Sicherheitsmethoden müssen Sie AWS Identity and Access Management (IAM)-Konten verwenden, um den Zugriff auf Amazon DocumentDB-API-Operationen zu steuern, insbesondere Operationen, die Amazon DocumentDB-Ressourcen erstellen, ändern oder löschen. Zu solchen Ressourcen gehören Cluster, Sicherheitsgruppen und Parametergruppen. Sie müssen IAM auch verwenden, um Aktionen zu steuern, die allgemeine administrative Aktionen wie das Sichern der Wiederherstellung von Clustern ausführen. Verwenden Sie beim Erstellen von IAM-Rollen das Prinzip der geringsten Berechtigung.

- Erzwingen Sie die geringste Berechtigung mit rollenbasierter Zugriffssteuerung.
- Weisen Sie jeder Person, die Amazon DocumentDB-Ressourcen verwaltet, ein individuelles IAM-Konto zu. Verwenden Sie nicht den AWS-Konto Root-Benutzer, um Amazon DocumentDB-Ressourcen zu verwalten. Erstellen Sie einen IAM-Benutzer für jede Person einschließlich Sie selbst.
- Gewähren Sie jedem IAM-Benutzer die Mindestberechtigungen, die für die Ausführung seiner Aufgaben erforderlich sind.
- Verwenden Sie IAM-Gruppen, um Berechtigungen für mehrere Benutzer effektiv zu verwalten. Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch. Weitere Informationen zu bewährten Methoden für IAM finden Sie unter Bewährte Methoden für IAM.

- Wechseln Sie regelmäßig die IAM-Anmeldeinformationen.
- Konfigurieren Sie AWS Secrets Manager so, dass die Secrets für Amazon DocumentDB automatisch rotiert werden. Weitere Informationen finden Sie unter Rotieren Ihrer AWS Secrets-Manager-Secrets und Rotieren von Secrets für Amazon DocumentDB im AWS Secrets-Manager-Benutzerhandbuch.
- Gewähren Sie jedem Amazon DocumentDB-Benutzer die Mindestberechtigungen, die für die Ausführung seiner Aufgaben erforderlich sind. Weitere Informationen finden Sie unter Datenbankzugriff mit rollenbasierter Zugriffskontrolle.
- Verwenden Sie Transport Layer Security (TLS), um Ihre Daten während der Übertragung und Ihre Daten im Ruhezustand AWS KMS zu verschlüsseln.

Kostenoptimierung

Die folgenden bewährten Methoden können Ihnen bei der Verwaltung und Minimierung Ihrer Kosten bei der Verwendung von Amazon DocumentDB helfen. Preisinformationen finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) – Preise und Häufig FAQs zu Amazon DocumentDB (mit MongoDB-Kompatibilität).

- Erstellen Sie Gebührenlimit-Warnungen für Schwellenwerte von 50 Prozent und 75 Prozent Ihrer erwarteten Rechnung für den Monat. Weitere Informationen zum Erstellen von Gebührenlimit-Warnungen finden Sie unter Erstellen einer Gebührenlimit-Warnung.
- Die Architektur von Amazon DocumentDB trennt Speicher und Rechenleistung, sodass auch ein Einzel-Instance-Cluster äußerst langlebig ist. Das Cluster-Speicher-Volume repliziert Daten auf sechs Arten in drei Availability Zones und bietet unabhängig von der Anzahl der Instances im Cluster eine extrem hohe Beständigkeit. Ein typischer Produktions-Cluster verfügt über mindestens drei Instances, um eine hohe Verfügbarkeit bereitzustellen. Sie können jedoch die Kosten optimieren, indem Sie einen einzelnen Instance-Entwicklungs-Cluster verwenden, wenn keine hohe Verfügbarkeit erforderlich ist.
- Halten Sie bei Entwicklungs- und Testszenarien einen Cluster an, wenn er nicht mehr benötigt wird, und starten Sie den Cluster, wenn die Entwicklung fortgesetzt wird. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.
- Sowohl für TTL als auch Change Streams fallen beim Schreiben, Lesen und Löschen von Daten E/As an. Wenn Sie diese Funktionen aktiviert haben, sie aber in Ihrer Anwendung nicht nutzen, kann die Deaktivierung der Funktionen zur Kostensenkung beitragen.

Verwendung von Metriken zur Identifizierung von Problemen mit der Leistung

Um Leistungsprobleme zu identifizieren, die durch unzureichende Ressourcen und andere häufig auftretende Engpässe verursacht werden, können Sie die für Ihren Amazon DocumentDB-Cluster verfügbaren Metriken überwachen.

Anzeigen von Leistungsmetriken

Überwachen Sie die Leistungsmetriken regelmäßig, um die Durchschnitts-, Höchst- und Mindestwerte für verschiedene Zeitbereiche anzuzeigen. Auf diese Weise können Sie feststellen, wenn die Leistung nachlässt. Sie können auch Amazon- CloudWatch Alarme für bestimmte Metrikschwellenwerte festlegen, sodass Sie benachrichtigt werden, wenn sie erreicht werden.

Um Probleme mit der Leistung zu beheben, müssen Sie die Basisleistung des Systems kennen. Nachdem Sie einen neuen Cluster eingerichtet haben und er mit einer typischen Workload ausgeführt wird, erfassen Sie die Durchschnitts-, Höchst- und Mindestwerte aller Leistungsmetriken in unterschiedlichen Intervallen (z. B. 1 Stunde, 24 Stunden, 1 Woche, 2 Wochen). Auf diese Weise erhalten Sie eine Vorstellung davon, was normal ist. Dies hilft, um Vergleichswerte für Betriebsstunden während und außerhalb von Spitzenbelastungen zu erhalten. Sie können diese Informationen anschließend verwenden, um festzustellen, wann die Leistung unter Standardwerte absinkt.

Sie können Leistungsmetriken mit der AWS Management Console oder anzeigen AWS CLI. Weitere Informationen finden Sie unter Wird angezeigt CloudWatch Daten.

Festlegen eines CloudWatch Alarms

Informationen zum Einstellen eines CloudWatch Alarms finden Sie unter Verwenden von Amazon-CloudWatch Alarmen im Amazon- CloudWatch Benutzerhandbuch.

Auswerten von Leistungsmetriken

Eine Instance besitzt mehrere unterschiedliche Kategorien von Metriken. Die Art und Weise, wie Sie bestimmen, welche Werte akzeptabel sind, ist von der Metrik abhängig.

CPU

- CPU-Auslastung – Der Prozentsatz der verwendeten Computerverarbeitungskapazität.

Arbeitsspeicher

- Freisetzbarer Speicher – Wie viel RAM auf der Instance verfügbar ist.
- Auslagerungsnutzung – Wie viel Auslagerungsspeicher von der Instance in Megabyte verwendet wird.

Eingabe-/Ausgabe-Operationen

- Lese-IOPS, Schreib-IOPS – Die durchschnittliche Anzahl von Festplattenlese- oder Schreibvorgängen pro Sekunde.
- Leselatenz, Schreiblatenz – Die durchschnittliche Zeit für einen Lese- oder Schreibvorgang in Millisekunden.
- Lesedurchsatz, Schreibdurchsatz – Die durchschnittliche Anzahl von Megabyte, die pro Sekunde von der Festplatte gelesen oder auf die Festplatte geschrieben werden.
- Tiefe der Festplattenwarteschlange – Die Anzahl der E/A-Operationen, die darauf warten, auf die Festplatte geschrieben oder von dieser gelesen zu werden.

Netzwerkdatenverkehr

- Netzwerkempfangsdurchsatz, Netzwerkübertragungsdurchsatz – Die Rate des Netzwerkverkehrs zur und von der Instance in Megabyte pro Sekunde.

Datenbankverbindungen

- DB-Verbindungen – Die Anzahl der Clientsitzungen, die mit der Instance verbunden sind.

Allgemein ausgedrückt, sind die zulässigen Werte für Leistungsmetriken davon abhängig, wie die Basisleistung aussieht und welche Aufgaben von Ihrer Anwendung ausgeführt werden. Prüfen Sie, ob dauerhafte oder tendenzielle Abweichungen von Ihrer Ausgangsbasis vorliegen.

Nachstehend folgen Empfehlungen und Ratschläge zu bestimmten Arten von Metriken:

- Hohe CPU-Auslastung – Hohe Werte für die CPU-Auslastung können angemessen sein, vorausgesetzt, sie entsprechen Ihren Zielen für Ihre Anwendung (z. B. Durchsatz oder Gleichzeitigkeit) und werden erwartet. Wenn die CPU-Auslastung konsistent mehr als 80 Prozent beträgt, sollten Sie eine Aufwärtsskalierung Ihrer Instances in Betracht ziehen.

- Hoher RAM-Verbrauch – Wenn Ihre `FreeableMemory` Metrik häufig unter 10 % des gesamten Instance-Speichers fällt, sollten Sie erwägen, Ihre Instances hochzuskalieren. Weitere Informationen darüber, was passiert, wenn Ihre DocumentDB-Instance unter hohem Speicherdruck steht, finden Sie unter Amazon DocumentDB Resource Governance.
- Auslagerungsnutzung – Diese Metrik sollte bei oder nahe Null bleiben. Wenn die Swap-Nutzung erheblich ist, sollten Sie eine Aufwärtsskalierung Ihrer Instances in Betracht ziehen.
- Netzwerkverkehr – Wenden Sie sich für Netzwerkverkehr an Ihren Systemadministrator, um zu erfahren, wie hoch der erwartete Durchsatz für Ihr Domänennetzwerk und Ihre Internetverbindung ist. Überprüfen Sie den Netzwerkdatenverkehr, wenn der Durchsatz dauerhaft unter dem erwarteten Wert liegt.
- Datenbankverbindungen – Erwägen Sie, Datenbankverbindungen einzuschränken, wenn Sie eine hohe Anzahl von Benutzerverbindungen zusammen mit einer Abnahme der Instance-Leistung und der Reaktionszeit feststellen. Die optimale Anzahl der Benutzerverbindungen für Ihre Instance ist von der Instance-Klasse und der Komplexität der ausgeführten Operationen abhängig. Bei Problemen mit Leistungsmetriken sollten Sie zunächst die am häufigsten verwendeten und kostspieligsten Abfragen anpassen, um festzustellen, ob dies den Druck auf die Systemressourcen verringert und die Leistung verbessert.

Wenn Ihre Abfragen optimiert sind und ein Problem weiterhin besteht, sollten Sie erwägen, Ihre Amazon DocumentDB-Instance-Klasse auf eine Klasse mit mehr Ressourcen (CPU, RAM, Festplattenspeicher, Netzwerkbandbreite, E/A-Kapazität) zu aktualisieren, die mit dem aufgetretenen Problem zusammenhängt.

Optimieren von Abfragen

Eine der besten Möglichkeiten zur Verbesserung der Cluster-Leistung besteht darin, die am häufigsten verwendeten und ressourcenintensivsten Abfragen so anzupassen, dass ihre Ausführung weniger aufwändig wird.

Mit dem Profiler (siehe Profilierung von Amazon DocumentDB-Vorgängen) können Sie die Ausführungszeit und Details der Vorgänge protokollieren, die für Ihren Cluster ausgeführt wurden. Profiler ist nützlich für die Überwachung der langsamsten Operationen in Ihrem Cluster, um die Leistung einzelner Abfragen und die allgemeine Cluster-Leistung zu verbessern.

Sie können den Befehl `explain` auch verwenden, um zu erfahren, wie ein Abfrageplan für eine bestimmte Abfrage analysiert wird. Mithilfe dieser Informationen können Sie eine Abfrage oder

zugrundeliegende Sammlung ändern, um die Abfrageleistung zu verbessern (z. B. Hinzufügen eines Indexes).

TTL- und Zeitreihen-Workloads

Das Löschen von Dokumenten, das aus dem Ablauf des TTL-Index resultiert, ist ein hochaufwendiges Verfahren. Es wird nicht garantiert, dass Dokumente innerhalb eines bestimmten Zeitraums gelöscht werden. Faktoren wie die Größe der Instance, die Ressourcenauslastung der Instance, die Dokumentgröße, der Gesamtdurchsatz, die Anzahl der Indizes und die Frage, ob die Indizes und der Arbeitssatz in den Speicher passen, können den Zeitpunkt beeinflussen, zu dem abgelaufene Dokumente durch den TTL-Prozess gelöscht werden.

Wenn der TTL-Monitor Ihre Dokumente löscht, entstehen bei jeder Löschung E/A-Kosten, was den Rechnungsbetrag erhöht. Wenn der Durchsatz und die TTL-Löschraten steigen, müssen Sie aufgrund der erhöhten E/A-Nutzung mit einer höheren Rechnung rechnen. Wenn Sie jedoch keinen TTL-Index erstellen, um Dokumente zu löschen, sondern Dokumente nach Zeit in Sammlungen segmentieren und diese Sammlungen einfach entfernen, wenn sie nicht mehr benötigt werden, fallen keine I/O-Kosten an. Dies kann deutlich kostengünstiger sein als die Verwendung eines TTL-Index.

Bei Zeitreihen-Workloads können Sie die Erstellung von fortlaufenden Sammlungen anstelle eines TTL-Index in Betracht ziehen, da fortlaufende Sammlungen eine leistungsfähigere Methode zum Löschen von Daten und weniger E/A-intensiv sein können. Wenn Sie große Sammlungen haben (insbesondere Sammlungen mit mehr als 1 TB) oder die E/A-Kosten für das TTL-Löschen ein Problem darstellen, empfiehlt es sich, Dokumente basierend auf der Zeit in Sammlungen zu partitionieren und Sammlungen zu löschen, wenn die Dokumente nicht mehr benötigt werden. Sie können eine Sammlung pro Tag oder eine Sammlung pro Woche erstellen, abhängig von der Datenaufnahmerate. Bei je nach Anwendung variierenden Anforderungen gilt als gute Faustregel, besser über mehr kleinere Sammlungen als über einige große Sammlungen zu verfügen. Das Löschen dieser Sammlungen verursacht keine E/A-Kosten und kann schneller und kostengünstiger sein als die Verwendung eines TTL-Index.

Migrationen

Als bewährte Methode empfehlen wir, dass Sie bei der Migration von Daten zu Amazon DocumentDB zunächst Ihre Indizes in Amazon DocumentDB erstellen, bevor Sie die Daten migrieren. Amazon DocumentDB Indizes zuerst zu erstellen, kann die Gesamtzeit verkürzen und die Geschwindigkeit der Migration erhöhen. Dazu können Sie das Amazon DocumentDB Index Tool verwenden. Weitere Informationen zu Migrationen finden Sie im Amazon DocumentDB-Migrationshandbuch.

Wir empfehlen außerdem, dass Sie Ihre Anwendung vor der Migration Ihrer Produktionsdatenbank vollständig auf Amazon DocumentDB testen, wobei Funktionalität, Leistung, Betrieb und Kosten berücksichtigt werden.

Arbeiten mit Cluster-Parametergruppen

Es wird empfohlen, Änderungen an Cluster-Parametergruppen zuerst an einem Test-Cluster auszuprobieren, bevor Sie die Änderungen auf Ihre Produktions-Cluster anwenden. Weitere Informationen zum Sichern Ihres Clusters finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.

Aggregation-Pipeline-Abfragen

Wenn Sie eine Aggregation-Pipeline-Abfrage mit mehreren Stufen erstellen und nur eine Teilmenge der Daten in der Abfrage auswerten, verwenden Sie die `$match`-Stufe als erste Stufe oder am Anfang der Pipeline. Wenn Sie `$match` zuerst verwenden, wird die Anzahl der Dokumente reduziert, die nachfolgende Stufen innerhalb der Aggregation-Pipeline-Abfrage verarbeiten müssen, wodurch die Leistung Ihrer Abfrage verbessert wird.

`batchInsert` und **`batchUpdate`**

Wenn Sie eine hohe Rate gleichzeitiger - `batchInsert` und/oder -`batchUpdate`Operationen ausführen und die Menge von `FreeableMemory` (CloudWatch Metrik) auf Ihrer primären Instance auf Null geht, können Sie entweder die Gleichzeitigkeit der Stapeleinfügung reduzieren oder den Workload aktualisieren oder, wenn die Gleichzeitigkeit des Workloads nicht reduziert werden kann, die Instance-Größe erhöhen, um die Menge von zu erhöhen`FreeableMemory`.

Funktionale Unterschiede: Amazon DocumentDB und MongoDB

Im Folgenden sind die funktionalen Unterschiede zwischen Amazon DocumentDB (mit MongoDB-Kompatibilität) und MongoDB aufgeführt.

Themen

Funktionsvorteile von Amazon DocumentDB

Implizite Transaktionen

In Amazon DocumentDB garantieren alle CRUD-Anweisungen (`findAndModify`, `update`, `insert`, `delete`) Atomizität und Konsistenz, auch für Operationen, die mehrere Dokumente ändern. Mit der Einführung von Amazon DocumentDB 4.0 werden jetzt explizite Transaktionen unterstützt, die ACID-Eigenschaften für Operationen mit mehreren Anweisungen und mehreren Sammlungen bereitstellen. Weitere Informationen zur Verwendung von Transaktionen in Amazon DocumentDB finden Sie unter Transaktionen.

Im Folgenden finden Sie Beispiele für Operationen in Amazon DocumentDB, die mehrere Dokumente ändern, die sowohl atomares als auch konsistentes Verhalten erfüllen.

```
db.miles.update(
    { "credit_card": { $eq: true } },
    { $mul: { "flight_miles.$[]": NumberInt(2) } },
    { multi: true }
)
```

```
db.miles.updateMany(
    { "credit_card": { $eq: true } },
    { $mul: { "flight_miles.$[]": NumberInt(2) } }
)
```

```
db.runCommand({
  update: "miles",
  updates: [
    {
      q: { "credit_card": { $eq: true } },
      u: { $mul: { "flight_miles.$[]": NumberInt(2) } },
      multi: true
    }
  ]
})
```

```
db.products.deleteMany({
  "cost": { $gt: 30.00 }
})
```

```
db.runCommand({
  delete: "products",
  deletes: [{ q: { "cost": { $gt: 30.00 } }, limit: 0 }]
})
```

Die einzelnen Operationen, aus denen die Massenoperationen bestehen, wie `updateMany` und `deleteMany`, sind atomar, aber die Gesamtheit der Massenoperation ist nicht atomar. Zum Beispiel ist die Gesamtheit der `insertMany`-Operation atomar, wenn die einzelnen Einfügeoperationen erfolgreich ohne Fehler ausgeführt werden. Wenn bei einer `insertMany` Operation ein Fehler auftritt, wird jede einzelne Einfügeanweisung innerhalb der `insertMany` Operation als atomare Operation ausgeführt. Wenn Sie ACID-Eigenschaften für `insertMany`-`updateMany`, - und -`deleteMany`Operationen benötigen, wird empfohlen, eine -Transaktion zu verwenden.

Aktualisierung bzgl. der Funktionsunterschiede

Amazon DocumentDB verbessert weiterhin die Kompatibilität mit MongoDB, indem es rückwärts von den Funktionen arbeitet, die unsere Kunden uns zur Entwicklung auffordern. Dieser Abschnitt enthält die funktionalen Unterschiede, die wir in Amazon DocumentDB entfernt haben, um Migrationen und die Erstellung von Anwendungen für unsere Kunden zu erleichtern.

Themen

Array-Indizierung

Ab dem 23. April 2020 unterstützt Amazon DocumentDB jetzt die Möglichkeit, Arrays zu indizieren, die größer als 2.048 Byte sind. Das Limit für ein einzelnes Element in einem Array bleibt weiterhin 2.048 Bytes, was mit MongoDB übereinstimmt.

Beim Erstellen eines neuen Index sind keine Maßnahmen erforderlich, um die Vorteile der verbesserten Funktionalität zu nutzen. Wenn Sie über einen vorhandenen Index verfügen, können Sie die verbesserte Funktionalität nutzen, indem Sie den Index löschen und anschließend neu erstellen. Die aktuelle Index-Version mit den verbesserten Fähigkeiten lautet `"v" : 3`.

> **Note**
>
> Bei Produktions-Clustern kann das Entfernen des Index Auswirkungen auf Ihre Anwendungsleistung haben. Wir empfehlen Ihnen, bei Änderungen an einem Produktionssystem zunächst Tests durchzuführen und dann mit Bedacht weiter vorzugehen. Darüber hinaus hängt die Zeit, die für die Neuerstellung des Index benötigt wird, von der Gesamtdatengröße der Sammlung ab.

Mit folgendem Befehl können Sie die Version Ihrer Indizes abfragen.

```
db.collection.getIndexes()
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus. In dieser Ausgabe ist die Indexversion `"v" : 3`, die die aktuellste Indexversion ist.

```
[
```

```
    {
        "v" : 3,
        "key" : {
        "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.test"
    }
]
```

Multikey-Indizes

Ab dem 23. April 2020 unterstützt Amazon DocumentDB jetzt die Möglichkeit, einen zusammengesetzten Index mit mehreren Schlüsseln im selben Array zu erstellen.

Beim Erstellen eines neuen Index sind keine Maßnahmen erforderlich, um die Vorteile der verbesserten Funktionalität zu nutzen. Wenn Sie über einen vorhandenen Index verfügen, können Sie die verbesserte Funktionalität nutzen, indem Sie den Index löschen und anschließend neu erstellen. Die aktuelle Index-Version mit den verbesserten Fähigkeiten lautet `"v" : 3`.

> **Note**
>
> Bei Produktions-Clustern kann das Entfernen des Index Auswirkungen auf Ihre Anwendungsleistung haben. Wir empfehlen Ihnen, bei Änderungen an einem Produktionssystem zunächst Tests durchzuführen und dann mit Bedacht weiter vorzugehen. Darüber hinaus hängt die Zeit, die für die Neuerstellung des Index benötigt wird, von der Gesamtdatengröße der Sammlung ab.

Mit folgendem Befehl können Sie die Version Ihrer Indizes abfragen.

```
db.collection.getIndexes()
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus. In dieser Ausgabe ist die Indexversion `"v" : 3`, die die aktuellste Indexversion ist.

```
[
    {
        "v" : 3,
```

```
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.test"
    }
]
```

Null-Zeichen in Zeichenfolgen

Ab dem 22. Juni 2020 unterstützt Amazon DocumentDB jetzt Nullzeichen (`'\0'`) in Zeichenfolgen.

Rollenbasierte Zugriffssteuerung

Ab dem 26. März 2020 unterstützt Amazon DocumentDB die rollenbasierte Zugriffskontrolle (RBAC) für integrierte Rollen. Weitere Informationen hierzu finden Sie unter Rollenbasierte Zugriffssteuerung.

`$regex`-Indizierung

Ab dem 22. Juni 2020 unterstützt Amazon DocumentDB jetzt die Möglichkeit für `$regex` Operatoren, einen Index zu verwenden.

Um einen Index mit dem `$regex`-Operator zu nutzen, müssen Sie den `hint()`-Befehl verwenden. Bei der Verwendung von `hint()` müssen Sie den Namen des Feldes angeben, auf dem Sie `$regex` anwenden möchten. Wenn Sie beispielsweise einen Index für Feld `product` mit dem Indexnamen `p_1` haben, nutzt `db.foo.find({product: /^x.*/}).hint({product:1})` den `p_1`-Index, aber `db.foo.find({product: /^x.*/}).hint("p_1")` nutzt den Index nicht. Sie können mit dem `explain()`-Befehl überprüfen, ob ein Index ausgewählt wird, oder indem Sie den Profiler zum Protokollieren langsamer Abfragen verwenden. Beispiel: `db.foo.find({product: /^x.*/}).hint("p_1").explain()`

> **Note**
>
> Die `hint()`-Methode kann nur mit jeweils einem Index verwendet werden.

Die Verwendung eines Index für eine `$regex`-Abfrage ist für regex-Abfragen optimiert, die ein Präfix verwenden und bei denen die regex-Optionen `I`, `m` oder `o` nicht angegeben werden.

Wenn Sie einen Index mit `$regex` verwenden, wird empfohlen, einen Index für hoch selektive Felder zu erstellen, bei denen die Anzahl der doppelten Werte weniger als 1 % der Gesamtzahl der Dokumente in der Sammlung beträgt. Wenn Ihre Sammlung beispielsweise 100.000 Dokumente enthält, erstellen Sie Indizes nur für Felder, in denen derselbe Wert 1.000 Mal oder weniger vorkommt.

Projektion für verschachtelte Dokumente

Es gibt einen funktionalen Unterschied `$project` zwischen Amazon DocumentDB und MongoDB in Version 3.6, der in Amazon DocumentDB 4.0 gelöst wurde, in Amazon DocumentDB 3.6 jedoch nicht unterstützt wird.

Amazon DocumentDB 3.6 berücksichtigt bei der Anwendung einer Projektion nur das erste Feld in einem verschachtelten Dokument, während MongoDB 3.6 Unterdokumente analysiert und die Projektion auch auf jedes Unterdokument anwendet.

Wenn die Projektion beispielsweise lautet`"a.b.c": 1`, funktioniert das Verhalten sowohl in Amazon DocumentDB als auch in MongoDB wie erwartet. Wenn die Projektion jedoch ist, wendet `{a:{b:{c:1}}}` Amazon DocumentDB 3.6 die Projektion nur auf `a` und nicht auf `b` oder `anc`. In Amazon DocumentDB 4.0 `{a:{b:{c:1}}}` wird die Projektion auf `a`, `b`und angewendet`c`.

Funktionale Unterschiede zu MongoDB

Themen

- $vectorSearch-Operator
- OpCountersCommand
- Admin-Datenbanken und Sammlungen
- cursormaxTimeMS
- explain()
- Einschränkungen für Feldnamen
- Indexerstellungen
- Suche mit leerem Schlüssel im Pfad
- MongoDB-APIs, -Operationen und -Datentypen
- mongodump - und mongorestore -Dienstprogramme
- Ergebnissortierung

- Wiederholbare Schreibvorgänge
- Sparse-Index
- Verwendung von $elemMatch innerhalb eines $all-Ausdrucks
- $ne, $nin, $nor, notexists, und $elemMatch Indizierung
- $lookup

`$vectorSearch`-Operator

Amazon DocumentDB unterstützt nicht `$vectorSearch` als unabhängigen Operator. Stattdessen unterstützen wir `vectorSearch` innerhalb des -`$search`Operators. Weitere Informationen finden Sie unter Vektorsuche nach Amazon DocumentDB.

`OpCountersCommand`

Das Verhalten von`OpCountersCommand` Amazon DocumentDB weicht von MongoDB `opcounters.command`wie folgt ab:

- MongoDB `opcounters.command` zählt alle Befehle außer Einfügen, Aktualisieren und Löschen, während Amazon DocumentDB`OpCountersCommand` ebenfalls den `find` Befehl ausschließt.
- Amazon DocumentDB zählt interne Befehle (z. B. `getCloudWatchMetricsV2`) zu `OpCountersCommand`.

Admin-Datenbanken und Sammlungen

Amazon DocumentDB unterstützt weder den Administrator oder die lokale Datenbank noch die MongoDB- `system.*` bzw. -`startup_log`Sammlungen.

`cursormaxTimeMS`

In Amazon DocumentDB setzt den Zähler für jede `getMore` Anforderung `cursor.maxTimeMS` zurück. Wenn also ein 3000MS angegeben `maxTimeMS` wird, nimmt die Abfrage 2800MS und jede nachfolgende `getMore` Anforderung 300MS in Anspruch, dann tritt für den Cursor kein Timeout auf. Der Cursor führt nur dann eine Zeitüberschreitung durch, wenn eine einzelne Operation, entweder die Abfrage oder eine einzelne `getMore` Anforderung, mehr benötigt als die angegebene `maxTimeMS`. Darüber hinaus wird der Timer, der die Cursor-Ausführungszeit überprüft, mit einer Granularität von fünf (5) Minuten ausgeführt.

explain()

Amazon DocumentDB emuliert die MongoDB 4.0-API auf einer speziell entwickelten Datenbank-Engine, die ein verteiltes, fehlertolerantes, selbstverstärkendes Speichersystem verwendet. Daher `explain()` können Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterschiedlich sein. Kunden, die die Kontrolle über ihren Abfrageplan wünschen, können den `$hint`-Operator verwenden, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Einschränkungen für Feldnamen

Amazon DocumentDB unterstützt keine Punkte „“. in einem Dokumentfeldnamen, z. B. `db.foo.insert({‘x.1’:1})`.

Amazon DocumentDB unterstützt auch nicht das Präfix $ in Feldnamen.

Versuchen Sie beispielsweise den folgenden Befehl in Amazon DocumentDB oder MongoDB :

```
rs0:PRIMARY< db.foo.insert({"a":{"$a":1}})
```

MongoDB gibt Folgendes zurück:

```
WriteResult({ "nInserted" : 1 })
```

Amazon DocumentDB gibt einen Fehler zurück:

```
WriteResult({
"nInserted" : 0,
"writeError" : {
"code" : 2,
"errmsg" : "Document can't have $ prefix field names: $a"
}
})
```

> **Note**
>
> Es gibt eine Ausnahme von diesem funktionalen Unterschied. Die folgenden Feldnamen, die mit dem Präfix $ beginnen, wurden auf die Whitelist gesetzt und können erfolgreich in Amazon DocumentDB verwendet werden: $id, $ref und $db.

Indexerstellungen

Amazon DocumentDB lässt zu, dass jeweils nur ein Index-Build für eine Sammlung ausgeführt wird. Entweder im Vordergrund oder im Hintergrund. Wenn Vorgänge wie `createIndex()` oder `dropIndex()` für dieselbe Sammlung auftreten, wenn gerade ein Index erstellt wird,, schlägt der neu versuchte Vorgang fehl.

Standardmäßig werden Index-Builds in Amazon DocumentDB und MongoDB Version 4.0 im Hintergrund ausgeführt. MongoDB Version 4.2 und höher ignoriert die Hintergrundindex-Build-Option, wenn createIndexes oder seine Shell-Hilfsprogramme `createIndex()` und angegeben sind`createIndexes()`.

Ein Time to Live (TTL)-Index beginnt mit dem Ablaufen von Dokumenten, nachdem der Indexaufbau abgeschlossen ist.

Suche mit leerem Schlüssel im Pfad

Wenn Sie nach einem Schlüssel suchen, der eine leere Zeichenfolge als Teil des Pfads enthält (z. B. `x.`, `x..b`), und das Objekt einen leeren Zeichenfolgenschlüsselpfad (z. B. `{"x" : [ { "" : 10 }, { "b" : 20 } ]}`) innerhalb eines Arrays hat, gibt Amazon DocumentDB andere Ergebnisse zurück, als wenn Sie dieselbe Suche in MongoDB ausführen würden.

In MongoDB funktioniert die leere Schlüsselpfadsuche innerhalb des Arrays wie erwartet, wenn sich der leere Zeichenfolgenschlüssel nicht am Ende der Pfadsuche befindet. Wenn sich der leere Zeichenfolgenschlüssel jedoch am Ende der Pfadsuche befindet, wird das Array nicht untersucht.

In Amazon DocumentDB wird jedoch nur das erste Element innerhalb des Arrays gelesen, da eine leere Zeichenfolge `getArrayIndexFromKeyString` in konvertiert`0`, sodass die Suche nach Zeichenfolgenschlüsseln als Suche nach Array-Index behandelt wird.

MongoDB-APIs, -Operationen und -Datentypen

Amazon DocumentDB ist mit den APIs MongoDB 3.6 und 4.0 kompatibel APIs. Eine up-to-date Liste der unterstützten Funktionen finden Sie unter Unterstützte MongoDB-APIs, -Operationen und -Datentypen.

`mongodump` - und **`mongorestore`** -Dienstprogramme

Amazon DocumentDB unterstützt keine Admin-Datenbank und verwirft oder stellt die Admin-Datenbank daher nicht wieder her, wenn die `mongodump`- oder -`mongorestore`Dienstprogramme

verwendet werden. Wenn Sie eine neue Datenbank in Amazon DocumentDB mit erstellen`mongorestore`, müssen Sie die Benutzerrollen zusätzlich zum Wiederherstellungsvorgang neu erstellen.

> **Note**
>
> Wir empfehlen MongoDB-Datenbanktools bis einschließlich Version 100.6.1 für Amazon DocumentDB . Sie können hier auf die Downloads der MongoDB-Datenbanktools zugreifenhttps://www.mongodb.com/download-center/database-tools/releases/archive.

Ergebnissortierung

Amazon DocumentDB garantiert keine implizite Ergebnissortierreihenfolge von Ergebnismengen. Um die Reihenfolge einer Ergebnismenge zu gewährleisten, geben Sie mit `sort()` explizit eine Sortierreihenfolge an.

Das folgende Beispiel sortiert die Artikel in der Inventur in absteigender Reihenfolge basierend auf dem Bestandsfeld.

```
db.inventory.find().sort({ stock: -1 })
```

Bei Verwendung der `$sort` Aggregationsstufe wird die Sortierreihenfolge nicht beibehalten, es sei denn, die `$sort` Stufe ist die letzte Phase in der Aggregationspipeline. Bei Verwendung der `$sort` Aggregationsstufe in Kombination mit der `$group` Aggregationsstufe wird die `$sort` Aggregationsstufe nur auf die - `$first` und -`$last`Aggregatoren angewendet. In Amazon DocumentDB 4.0 wurde Unterstützung für hinzugefügt, `$push` um die Sortierreihenfolge aus der vorherigen `$sort` Phase zu berücksichtigen.

Wiederholbare Schreibvorgänge

Beginnend mit MongoDB 4.2-kompatiblen Treibern sind wiederholbare Schreibvorgänge standardmäßig aktiviert. Amazon DocumentDB unterstützt derzeit jedoch keine wiederholbaren Schreibvorgänge. Der funktionale Unterschied wird sich in einer Fehlermeldung ähnlich der folgenden zeigen.

```
{"ok":0,"errmsg":"Unrecognized field: 'txnNumber'","code":9,"name":"MongoError"}
```

Wiederholbare Schreibvorgänge können über die Verbindungszeichenfolge (z. B. `MongoClient("mongodb://my.mongodb.cluster/db?retryWrites=false")`) oder das Schlüsselwortargument des MongoClient Konstruktors (z. B.) deaktiviert werden`MongoClient("mongodb://my.mongodb.cluster/db", retryWrites=False)`.

Im Folgenden finden Sie ein Python-Beispiel, das wiederholbare Schreibvorgänge in der Verbindungszeichenfolge deaktiviert.

```
client =
 pymongo.MongoClient('mongodb://
<username>:<password>@docdb-2019-03-17-16-49-12.cluster-ccuszbx3pn5e.us-
east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0',w='majority',j=True,retryWrites=False)
```

Sparse-Index

Um einen Sparse-Index zu verwenden, den Sie in einer Abfrage erstellt haben, müssen Sie die `$exists`-Bedingungen für die Felder verwenden, die der Index abdecken. Wenn Sie weglassen`$exists`, verwendet Amazon DocumentDB den Sparse-Index nicht.

Im Folgenden wird ein Beispiel gezeigt.

```
db.inventory.count({ "stock": { $exists: true }})
```

Für spärliche Indizes mit mehreren Schlüsseln unterstützt Amazon DocumentDB keine eindeutige Schlüsseleinschränkung, wenn die Suche nach einem Dokument zu einer Reihe von Werten führt und nur eine Teilmenge der indizierten Felder fehlt. Beispielsweise wird `createIndex({"a.b" : 1 }, { unique : true, sparse :true })` nicht unterstützt, wenn `"a" : [ { "b" : 2 }, { "c" : 1 } ]` eingegeben wird, da `"a.c"` im Index gespeichert ist.

Verwendung von $elemMatch innerhalb eines $all-Ausdrucks

Amazon DocumentDB unterstützt derzeit nicht die Verwendung des `$elemMatch` Operators innerhalb eines `$all` Ausdrucks. Sie können dies umgehen, indem Sie den Operator `$and` wie folgt mit `$elemMatch` verwenden.

Ursprüngliche Operation:

```
db.col.find({
```

```
  qty: {
    $all: [
      { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } },
      { "$elemMatch": { num: 40, size: "XL" } }
    ]
  }
})
```

Aktualisierte Operation:

```
db.col.find({
  $and: [
    { qty: { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } } },
    { qty: { "$elemMatch": { qty: 40, size: "XL" } } }
  ]
})
```

$ne, **$nin**, **$nor**, **$not$exists**, und **$elemMatch** Indizierung

Amazon DocumentDB unterstützt derzeit nicht die Möglichkeit, Indizes mit den `$distinct` Operatoren `$ne`, `$nin`, `$nor`, `$not$exists`, und zu verwenden. Daher führt die Verwendung dieser Operatoren zu Sammlungsscans. Die Durchführung eines Filters oder einer Übereinstimmung, bevor Sie einen dieser Operatoren verwenden, reduziert die Datenmenge, die gescannt werden muss, und kann so die Leistung verbessern.

Amazon DocumentDB hat Unterstützung für Index-Scans mit dem -`$elemMatch`Operator in Amazon DocumentDB 5.0 und elastischen Clustern hinzugefügt. Indexscans werden unterstützt, wenn der Abfragefilter nur eine Filterebene hat`$elemMatch`, aber nicht, wenn eine verschachtelte `$elemMatch` Abfrage enthalten ist.

`$elemMatch` Abfrageform, die Index-Scans in Amazon DocumentDB 5.0 unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

`$elemMatch` -Abfrageform, die keine Index-Scans in Amazon DocumentDB 5.0 unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e": "abc"} }} } })
```

$lookup

Amazon DocumentDB unterstützt die Möglichkeit, Gleichheitsübereinstimmungen durchzuführen (z. B. Left Outer Join), unterstützt jedoch keine unkorrelierten Unterabfragen.

Verwenden eines Index mit $lookup

Sie können jetzt einen Index mit dem `$lookup` Stufenoperator verwenden. Je nach Anwendungsfall gibt es mehrere Indizierungsalgorithmen, mit denen Sie die Leistung optimieren können. In diesem Abschnitt werden die verschiedenen Indizierungsalgorithmen für erläutert `$lookup` und Sie können den besten für Ihre Workload auswählen.

Standardmäßig verwendet Amazon DocumentDB den Hash-Algorithmus, wenn verwendet `allowDiskUse:false` wird, und die Sortierungszusammenführung, wenn verwendet `allowDiskUse:true` wird. In einigen Anwendungsfällen kann es sinnvoll sein, den Abfrageoptimierer zu zwingen, einen anderen Algorithmus zu verwenden. Im Folgenden finden Sie die verschiedenen Indizierungsalgorithmen, die der `$lookup` Aggregationsoperator verwenden kann:

- Verschachtelte Schleife: Ein verschachtelter Schleifenplan ist in der Regel für einen Workload von Vorteil, wenn die Fremdsammlung <1 GB ist und das Feld in der Fremdsammlung einen Index hat. Wenn der verschachtelte Loop-Algorithmus verwendet wird, zeigt der Erläuterungsplan die Stufe als an`NESTED_LOOP_LOOKUP`.
- Zusammenführung sortieren: Ein Zusammenführungsplan ist in der Regel für einen Workload von Vorteil, wenn die Fremdsammlung keinen Index für das Feld hat, das bei der Suche verwendet wird, und der Arbeitsdatensatz nicht in den Speicher passt. Wenn der Sortierzusammenführungsalgorithmus verwendet wird, zeigt der Erläuterungsplan die Stufe als an`SORT_LOOKUP`.
- Hash: Ein Hash-Plan ist in der Regel für einen Workload von Vorteil, wenn die Fremdsammlung < 1GB ist und der Arbeitsdatensatz in den Speicher passt. Wenn der Hash-Algorithmus verwendet wird, zeigt der Erläuterungsplan die Stufe als an`HASH_LOOKUP`.

Sie können den Indizierungsalgorithmus identifizieren, der für den `$lookup` Operator verwendet wird, indem Sie explain für die Abfrage verwenden. Im Folgenden finden Sie ein Beispiel.

```
db.localCollection.explain().
aggregate( [

   {
```

```
          $lookup:
                  {
                            from: "foreignCollection",
                            localField: "a",
       foreignField: "b",
       as: "joined"
}
   }
]

output
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "test.localCollection",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "SORT_AGGREGATE",
                "inputStage" : {
                    "stage" : "SORT",
                    "inputStage" : {
                        "stage" : "NESTED_LOOP_LOOKUP",
                        "inputStages" : [
                            {
                                "stage" : "COLLSCAN"
                            },
                            {
                                "stage" : "FETCH",
                                "inputStage" : {
                                    "stage" : "COLLSCAN"
                                }
                            }
                        ]
                    }
                }
            }
        }
    },
    "serverInfo" : {
        "host" : "devbox-test",
        "port" : 27317,
        "version" : "3.6.0"
    },
```

```
    "ok" : 1
}
```

Alternativ zur Verwendung der -`explain()`Methode können Sie den Profiler verwenden, um den Algorithmus zu überprüfen, der bei Ihrer Verwendung des -`$lookup`Operators verwendet wird. Weitere Informationen zum Profiler finden Sie unter Profilierung von Amazon DocumentDB-Vorgängen.

Verwenden eines **`planHint`**

Wenn Sie den Abfrageoptimierer zwingen möchten, einen anderen Indizierungsalgorithmus mit zu verwenden`$lookup`, können Sie einen verwenden`planHint`. Verwenden Sie dazu den Kommentar in den Optionen der Aggregationsphase, um einen anderen Plan zu erzwingen. Im Folgenden finden Sie ein Beispiel für die Syntax für den Kommentar:

```
comment : {
    comment :  “<string>”,
    lookupStage : { planHint : “SORT” | “HASH” | "NESTED_LOOP" }
}
```

Im Folgenden finden Sie ein Beispiel für die Verwendung der `planHint`, um den Abfrageoptimierer zur Verwendung des HASH Indizierungsalgorithmus zu zwingen:

```
db.foo.aggregate(
  [
    {
        $lookup:
        {
            from: "foo",
            localField: "_id",
            foreignField: "_id",
            as: "joined"
        },
    }
  ],
  {
    comment : "{ \\"lookupStage\\" : { \\"planHint\\": \\"HASH\\" }}"
```

Um zu testen, welcher Algorithmus für Ihre Workload am besten geeignet ist, können Sie den `executionStats` Parameter der `explain` Methode verwenden, um die Ausführungszeit

der `$lookup` Stufe zu messen und dabei den Indizierungsalgorithmus zu ändern (d. h. `HASH`/`SORT`/`NESTED_LOOP`).

Das folgende Beispiel zeigt, wie Sie verwenden, `executionStats` um die Ausführungszeit der `$lookup` Stufe mit dem `SORT` Algorithmus zu messen.

```
db.foo.explain(“executionStats”).aggregate(
  [
    {
        $lookup:
        {
            from: "foo",
            localField: "_id",
            foreignField: "_id",
            as: "joined"
        },
    }
  ],
  {
    comment : "{ \\"lookupStage\\" : { \\"planHint\": \\"SORT\\" }}"
```

Unterstützte MongoDB-APIs, -Operationen und -Datentypen

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein schneller, skalierbarer, hochverfügbarer und vollständig verwalteter Dokumentdatenbankservice, der MongoDB-Workloads unterstützt. Amazon DocumentDB ist mit den APIs MongoDB 3.6, 4.0 und 5.0 kompatibel. In diesem Abschnitt werden die unterstützten Funktionalitäten aufgeführt. Informationen zur Unterstützung von MongoDB-APIs und -Treibern finden Sie in den MongoDB-Community-Foren. Wenn Sie Unterstützung bei der Verwendung des Amazon DocumentDB-Service benötigen, wenden Sie sich bitte an das entsprechende AWS Support-Team. Funktionsunterschiede zwischen Amazon DocumentDB und MongoDB finden Sie unter Funktionale Unterschiede: Amazon DocumentDB und MongoDB.

MongoDB-Befehle und -Operatoren, die nur intern oder nicht auf einen verwalteten Service anwendbar sind, werden nicht unterstützt und sind nicht in der Liste der unterstützten Funktionalität enthalten.

Seit der Markteinführung haben wir mehr als 50 zusätzliche Funktionen hinzugefügt und werden uns weiter an den Anforderungen unserer Kunden ausrichten, um ihnen die benötigten Funktionen bereitzustellen. Informationen zu den letzten Starts finden Sie unter Amazon DocumentDB-Ankündigungen.

Wenn es eine Funktion gibt, die nicht unterstützt wird, die wir erstellen möchten, teilen Sie uns dies mit, indem Sie eine E-Mail mit Ihrer accountID, den angeforderten Funktionen und dem Anwendungsfall an das Amazon DocumentDB-Serviceteam senden.

Themen

Datenbank-Befehle

Themen

Administrative Befehle

Befehl	3.6	4,0	5.0	Elastic Cluster
Gedeckelte Sammlungen	Nein	Nein	Nein	Nein
cloneCollectionAsLimitiert	Nein	Nein	Nein	Nein
collMod	Teilweise	Teilweise	Teilweise	Ja
collMod expireAfterSeconds:	Ja	Ja	Ja	Nein
convertToCapped	Nein	Nein	Nein	Nein
copydb	Nein	Nein	Nein	Nein
create	Ja	Ja	Ja	Ja
createView	Nein	Nein	Nein	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
createIndexes	Ja	Ja	Ja	Ja
currentOp	Ja	Ja	Ja	Ja
fallen lassen	Ja	Ja	Ja	Ja
dropDatabase	Ja	Ja	Ja	Ja
dropIndexes	Ja	Ja	Ja	Ja
filemd5	Nein	Nein	Nein	Nein
killCursors	Ja	Ja	Ja	Ja
killOp	Ja	Ja	Ja	Ja
listCollections*	Ja	Ja	Ja	Ja
listDatabases	Ja	Ja	Ja	Ja
listIndexes	Ja	Ja	Ja	Ja
reIndex	Nein	Nein	Nein	Nein
renameCollection	Ja	Ja	Ja	Nein

* Der `type` Schlüssel in der Filteroption wird nicht unterstützt.

Aggregation

Befehl	3.6	4,0	5.0	Elastic Cluster
aggregate	Ja	Ja	Ja	Ja
count	Ja	Ja	Ja	Ja
distinct	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
mapReduce	Nein	Nein	Nein	Nein

Authentifizierung

Befehl	3.6	4,0	5.0	Elastic Cluster
authenticate	Ja	Ja	Ja	Ja
logout	Ja	Ja	Ja	Ja

Diagnose-Befehle

Befehl	3.6	4,0	5.0	Elastic Cluster
buildInfo	Ja	Ja	Ja	Ja
collStats	Ja	Ja	Ja	Ja
connPoolStats	Nein	Nein	Nein	Nein
connectionStatus	Ja	Ja	Ja	Ja
dataSize	Ja	Ja	Ja	Ja
dbHash	Nein	Nein	Nein	Nein
dbStats	Ja	Ja	Ja	Ja
explain	Ja	Ja	Ja	Ja
Erklären: executionStats	Ja	Ja	Ja	Ja
features	Nein	Nein	Nein	Nein
hostInfo	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
listCommands	Ja	Ja	Ja	Ja
Profiler	Ja	Ja	Ja	Nein
serverStatus	Ja	Ja	Ja	Ja
top	Ja	Ja	Ja	Ja

Abfrage- und Schreiboperationen

Befehl	3.6	4,0	5.0	Elastic Cluster
delete	Ja	Ja	Ja	Ja
find	Ja	Ja	Ja	Ja
findAndModify	Ja	Ja	Ja	Ja
getLastError	Nein	Nein	Nein	Nein
getMore	Ja	Ja	Ja	Ja
getPrevError	Nein	Nein	Nein	Nein
insert	Ja	Ja	Ja	Ja
parallelCollectionScan	Nein	Nein	Nein	Nein
resetError	Nein	Nein	Nein	Nein
update	Ja	Ja	Ja	Ja
Change streams	Ja	Ja	Ja	Nein
GridFS	Nein	Nein	Nein	Nein
ReplaceOne	Ja	Ja	Ja	Ja

Befehle zur Rollenverwaltung

E	3	4	5	E C
c e	Ja	Ja	Ja	Nein
d	Ja	Ja	Ja	Nein
c le ta	Ja	Ja	Ja	Nein a
g s	Ja	Ja	Ja	Nein
r v c	Ja	Ja	Ja	Nein
r e le	Ja	Ja	Ja	Nein l
u e	Ja	Ja	Ja	Nein
r	Ja	Ja	Ja	Nein

Sitzungsbefehle

Befehl	3.6	4,0	5.0	Elastic Cluster
abortTransaction	Nein	Ja	Ja	Nein
commitTra nsaction	Nein	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
endSessions	Nein	Nein	Nein	Nein
killAllSessions	Nein	Ja	Ja	Nein
killAllSessionsByPattern	Nein	Nein	Nein	Nein
killSessions	Nein	Ja	Ja	Nein
refreshSessions	Nein	Nein	Nein	Nein
startSession	Nein	Ja	Ja	Nein

Benutzerverwaltung

Befehl	3.6	4,0	5.0	Elastic Cluster
createUser	Ja	Ja	Ja	Ja
dropAllUsersFromDatabase	Ja	Ja	Ja	Ja
dropUser	Ja	Ja	Ja	Ja
grantRolesToBenutzer	Ja	Ja	Ja	Ja
revokeRolesFromBenutzer	Ja	Ja	Ja	Ja
updateUser	Ja	Ja	Ja	Ja
userInfo	Ja	Ja	Ja	Ja

Sharding-Befehle

Befehl	Elastic Cluster
abortReshardCollection	Nein
addShard	Nein
addShardToZone	Nein
balancerCollectionStatus	Nein
balancerStart	Nein
balancerStatus	Nein
balancerStop	Nein
checkShardingIndex	Nein
clearJumboFlag	Nein
cleanupOrphaned	Nein
cleanupReshardCollection	Nein
commitReshardCollection	Nein
enableSharding	Ja
flushRouterConfig	Nein
getShardMap	Nein
getShardVersion	Nein
isdbgrid	Nein
listShards	Nein
medianKey	Nein

Befehl	Elastic Cluster
moveChunk	Nein
movePrimary	Nein
mergeChunks	Nein
refineCollectionShardSchlüssel	Nein
removeShard	Nein
removeShardFromZone	Nein
reshardCollection	Nein
setAllowMigrations	Nein
setShardVersion	Nein
shardCollection	Ja
shardingState	Nein
split	Nein
splitVector	Nein
unsetSharding	Nein
updateZoneKeyBereich	Nein

Abfrage- und Projektions-Operatoren

Themen

- Element-Operatoren
- Auswertungsabfrage-Operatoren
- Logische Operatoren
- Projektions-Operatoren

Array-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$all	Ja	Ja	Ja	Ja
$elemMatch	Ja	Ja	Ja	Ja
$size	Ja	Ja	Ja	Ja

Bitwise-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$bitsAllSet	Ja	Ja	Ja	Ja
$bitsAnySet	Ja	Ja	Ja	Ja
$bitsAllClear	Ja	Ja	Ja	Ja
$bitsAnyClear	Ja	Ja	Ja	Ja

Kommentar-Operator

Befehl	3.6	4,0	5.0	Elastic Cluster
$comment	Ja	Ja	Ja	Ja

Vergleichsoperatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$eq	Ja	Ja	Ja	Ja
$gt	Ja	Ja	Ja	Ja
$gte	Ja	Ja	Ja	Ja
$lt	Ja	Ja	Ja	Ja
$lte	Ja	Ja	Ja	Ja
$ne	Ja	Ja	Ja	Ja
$in	Ja	Ja	Ja	Ja
$nin	Ja	Ja	Ja	Ja

Element-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$exists	Ja	Ja	Ja	Ja
$type	Ja	Ja	Ja	Ja

Auswertungsabfrage-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$expr	Nein	Nein	Nein	Nein
$jsonSchema	Nein	Ja	Ja	Nein
$mod	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
$regex	Ja	Ja	Ja	Ja
$text	Nein	Nein	Ja	Nein
$where	Nein	Nein	Nein	Nein

Logische Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$or	Ja	Ja	Ja	Ja
$and	Ja	Ja	Ja	Ja
$not	Ja	Ja	Ja	Ja
$nor	Ja	Ja	Ja	Ja

Projektions-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$	Ja	Ja	Ja	Ja
$elemMatch	Ja	Ja	Ja	Ja
$meta	Nein	Nein	Ja	Nein
$slice	Ja	Ja	Ja	Ja

Update-Operatoren

Themen

- Array-Operatoren

Array-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$	Ja	Ja	Ja	Ja
$[]	Ja	Ja	Ja	Ja
$[<identifier>]	Ja	Ja	Ja	Ja
$addToSet	Ja	Ja	Ja	Ja
$pop	Ja	Ja	Ja	Ja
$pullAll	Ja	Ja	Ja	Ja
$pull	Ja	Ja	Ja	Ja
$push	Ja	Ja	Ja	Ja

Bitwise-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$bit	Ja	Ja	Ja	Ja

Feld-Operatoren

Operator	3.6	4,0	5.0	Elastic Cluster
$inc	Ja	Ja	Ja	Ja

Operator	3.6	4,0	5.0	Elastic Cluster
$mul	Ja	Ja	Ja	Ja
$rename	Ja	Ja	Ja	Ja
$setOnInsert	Ja	Ja	Ja	Ja
$set	Ja	Ja	Ja	Ja
$unset	Ja	Ja	Ja	Ja
$min	Ja	Ja	Ja	Ja
$max	Ja	Ja	Ja	Ja
$currentDate	Ja	Ja	Ja	Ja

Update-Modifikatoren

Operator	3.6	4,0	5.0	Elastic Cluster
$each	Ja	Ja	Ja	Ja
$slice	Ja	Ja	Ja	Ja
$sort	Ja	Ja	Ja	Ja
$position	Ja	Ja	Ja	Ja

Geodaten

Geometry-Spezifizierer

Abfrageauswahl	3.6	4,0	5.0	Elastic Cluster
$box	Nein	Nein	Nein	Nein

Abfrageauswahl	3.6	4,0	5.0	Elastic Cluster
$center	Nein	Nein	Nein	Nein
$centerSphere	Nein	Nein	Nein	Nein
$nearSphere	Ja	Ja	Ja	Nein
$geometry	Ja	Ja	Ja	Nein
$maxDistance	Ja	Ja	Ja	Nein
$minDistance	Ja	Ja	Ja	Nein
$polygon	Nein	Nein	Nein	Nein
$uniqueDocs	Nein	Nein	Nein	Nein

Abfrageauswahl

Befehl	3.6	4,0	5.0	Elastic Cluster
$geoIntersects	Ja	Ja	Ja	Nein
$geoWithin	Ja	Ja	Ja	Nein
$near	Nein	Nein	Nein	Nein
$nearSphere	Ja	Ja	Ja	Nein
$polygon	Nein	Nein	Nein	Nein
$uniqueDocs	Nein	Nein	Nein	Nein

Cursor-Methoden

Befehl	3.6	4,0	5.0	Elastic Cluster
cursor.batchSize()	Ja	Ja	Ja	Ja
cursor.close()	Ja	Ja	Ja	Ja
cursor.isClosed()	Ja	Ja	Ja	Ja
cursor.collation()	Nein	Nein	Nein	Nein
cursor.comment()	Ja	Ja	Ja	Ja
cursor.count()	Ja	Ja	Ja	Ja
cursor.explain()	Ja	Ja	Ja	Nein
cursor.forEach()	Ja	Ja	Ja	Ja
cursor.hasNext()	Ja	Ja	Ja	Ja
cursor.hint()	Ja	Ja	Ja	Ja*
cursor.isExhausted()	Ja	Ja	Ja	Nein
cursor.itcount()	Ja	Ja	Ja	Nein
cursor.limit()	Ja	Ja	Ja	Nein
cursor.map()	Ja	Ja	Ja	Nein
cursor.maxScan()	Ja	Ja	Ja	Nein
cursor.maxTimeMS()	Ja	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
cursor.max()	Nein	Nein	Nein	Nein
cursor.min()	Nein	Nein	Nein	Nein
cursor.next()	Ja	Ja	Ja	Ja
Cursor.noCursorTimeout()	Nein	Nein	Nein	Nein
Cursor.objsLeftInBatch()	Ja	Ja	Ja	Nein
cursor.pretty()	Ja	Ja	Ja	Nein
cursor.readConcern()	Ja	Ja	Ja	Nein
cursor.readPref()	Ja	Ja	Ja	Nein
cursor.returnKey()	Nein	Nein	Nein	Nein
Cursor.showRecordId()	Nein	Nein	Nein	Nein
cursor.size()	Ja	Ja	Ja	Nein
cursor.skip()	Ja	Ja	Ja	Nein
cursor.sort()	Ja	Ja	Ja	Nein
cursor.tailable()	Nein	Nein	Nein	Nein
cursor.toArray()	Ja	Ja	Ja	Nein

* Index `hint` wird mit Indexausdrücken unterstützt. Beispiel: `db.foo.find().hint({x:1})`

Aggregation-Pipeline-Operatoren

Themen

Akkumulator-Ausdrücke

Expression	3.6	4,0	5.0	Elastic Cluster
$sum	Ja	Ja	Ja	Ja
$avg	Ja	Ja	Ja	Ja

Expression	3.6	4,0	5.0	Elastic Cluster
$first	Ja	Ja	Ja	Ja
$last	Ja	Ja	Ja	Ja
$max	Ja	Ja	Ja	Ja
$min	Ja	Ja	Ja	Ja
$push	Ja	Ja	Ja	Ja
$addToSet	Ja	Ja	Ja	Ja
$stdDevPop	Nein	Nein	Nein	Ja
$stdDevSamp	Nein	Nein	Nein	Ja
$accumulator	-	-	Nein	Nein
$count	-	-	Nein	Nein

Arithmetische Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$abs	Ja	Ja	Ja	Ja
$add	Ja	Ja	Ja	Ja
$ceil	Nein	Ja	Ja	Nein
$divide	Ja	Ja	Ja	Ja
$exp	Nein	Ja	Ja	Nein
$floor	Nein	Ja	Ja	Nein
$ln	Nein	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
$log	Nein	Ja	Ja	Nein
$log10	Nein	Ja	Ja	Nein
$mod	Ja	Ja	Ja	Ja
$multiply	Ja	Ja	Ja	Ja
$pow	Nein	Nein	Nein	Nein
$sqrt	Nein	Ja	Ja	Nein
$subtract	Ja	Ja	Ja	Ja
$trunc	Nein	Nein	Nein	Nein
$ rund	-	-	Nein	Nein

Array-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$arrayElemAt	Ja	Ja	Ja	Ja
$arrayToObject	Ja	Ja	Ja	Ja
$concatArrays	Ja	Ja	Ja	Ja
$filter	Ja	Ja	Ja	Ja
$indexOfArray	Ja	Ja	Ja	Ja
$isArray	Ja	Ja	Ja	Ja
$objectToArray	Ja	Ja	Ja	Ja
$range	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
$reverseArray	Ja	Ja	Ja	Ja
$reduce	Ja	Ja	Ja	Ja
$size	Ja	Ja	Ja	Ja
$slice	Ja	Ja	Ja	Ja
$zip	Ja	Ja	Ja	Ja
$in	Ja	Ja	Ja	Ja
$first	-	-	Nein	Nein
$last	-	-	Nein	Nein

Boolesche Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$and	Ja	Ja	Ja	Ja
$or	Ja	Ja	Ja	Ja
$not	Ja	Ja	Ja	Ja

Vergleichsoperatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$cmp	Ja	Ja	Ja	Ja
$eq	Ja	Ja	Ja	Ja
$gt	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
$gte	Ja	Ja	Ja	Ja
$lt	Ja	Ja	Ja	Ja
$lte	Ja	Ja	Ja	Ja
$ne	Ja	Ja	Ja	Ja

Operatoren für bedingte Ausdrücke

Befehl	3.6	4,0	5.0	Elastic Cluster
$cond	Ja	Ja	Ja	Ja
$ifNull	Ja	Ja	Ja	Ja
$switch	Nein	Nein	Nein	Nein

Datentyp-Operator

Befehl	3.6	4,0	5.0	Elastic Cluster
$type	Ja	Ja	Ja	Ja

Datengrößenoperator

Befehl	3.6	4,0	5.0	Elastic Cluster
$binarySize	-	-	Nein	Nein
$bsonSize	-	-	Nein	Nein

Datums-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$dateAdd	Nein	Nein	Ja	Ja
$dateSubtract	Nein	Nein	Ja	Ja
$dayOfYear	Ja	Ja	Ja	Ja
$dayOfMonth	Ja	Ja	Ja	Ja
$dayOfWeek	Ja	Ja	Ja	Ja
$year	Ja	Ja	Ja	Ja
$month	Ja	Ja	Ja	Ja
$week	Ja	Ja	Ja	Ja
$hour	Ja	Ja	Ja	Ja
$minute	Ja	Ja	Ja	Ja
$second	Ja	Ja	Ja	Ja
$millisecond	Ja	Ja	Ja	Ja
$dateToString	Ja	Ja	Ja	Ja
$isoDayOf Woche	Ja	Ja	Ja	Ja
$isoWeek	Ja	Ja	Ja	Ja
$dateFromParts	Nein	Nein	Nein	Nein
$dateToParts	Nein	Nein	Nein	Nein
$dateFromString	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
$isoWeekYear	Ja	Ja	Ja	Ja
$dataTrunc	-	-	Nein	Nein
$dataDiff	-	-	Nein	Nein

Literal-Operator

Befehl	3.6	4,0	5.0	Elastic Cluster
$literal	Ja	Ja	Ja	Ja

Merge-Operator

Befehl	3.6	4,0	5.0	Elastic Cluster
$mergeObjects	Ja	Ja	Ja	Ja

Naturischer Operator

Befehl	3.6	4,0	5.0	Elastic Cluster
$ natürlich	Ja	Ja	Ja	Ja

Satzoperatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$setEquals	Ja	Ja	Ja	Ja
$setIntersection	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
$setUnion	Ja	Ja	Ja	Ja
$setDifference	Nein	Ja	Ja	Ja
$setIsSubset	Ja	Ja	Ja	Ja
$anyElementTrue	Nein	Ja	Ja	Ja
$allElementsTrue	Nein	Ja	Ja	Ja

Stage-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$collStats	Nein	Nein	Nein	Nein
$project	Ja	Ja	Ja	Ja
$match	Ja	Ja	Ja	Ja
$redact	Ja	Ja	Ja	Ja
$limit	Ja	Ja	Ja	Ja
$skip	Ja	Ja	Ja	Ja
$unwind	Ja	Ja	Ja	Ja
$group	Ja	Ja	Ja	Ja
$sample	Ja	Ja	Ja	Nein
$sort	Ja	Ja	Ja	Ja
$geoNear	Ja	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
$lookup	Ja	Ja	Ja	Ja
$out	Ja	Ja	Ja	Nein
$indexStats	Ja	Ja	Ja	Ja
$facet	Nein	Nein	Nein	Nein
$bucket	Nein	Nein	Nein	Nein
$bucketAuto	Nein	Nein	Nein	Nein
$sortByCount	Nein	Nein	Nein	Nein
$addFields	Ja	Ja	Ja	Ja
$replaceRoot	Ja	Ja	Ja	Ja
$count	Ja	Ja	Ja	Ja
$currentOp	Ja	Ja	Ja	Nein
$listLocalSessions	Nein	Nein	Nein	Nein
$listSessions	Nein	Nein	Nein	Nein
$graphLookup	Nein	Nein	Nein	Nein
$merge	-	-	Nein	Nein
$planCacheStats	-	-	Nein	Nein
$setWindowFields	-	-	Nein	Nein
$unionWith	-	-	Nein	Nein
$unset	-	-	Nein	Nein

Zeichenfolgen-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$concat	Ja	Ja	Ja	Ja
$indexOfBytes	Ja	Ja	Ja	Ja
$indexOfCP	Ja	Ja	Ja	Ja
$ltrim	Nein	Nein	Nein	Nein
$rtrim	Nein	Nein	Nein	Nein
$split	Ja	Ja	Ja	Ja
$strcasecmp	Ja	Ja	Ja	Ja
$strLenBytes	Ja	Ja	Ja	Ja
$strLenCP	Ja	Ja	Ja	Ja
$substr	Ja	Ja	Ja	Ja
$substrBytes	Ja	Ja	Ja	Ja
$substrCP	Ja	Ja	Ja	Ja
$toLower	Ja	Ja	Ja	Ja
$toUpper	Ja	Ja	Ja	Ja
$trim	Nein	Nein	Nein	Nein
$regxFind	-	-	Nein	Nein
$regexFindAll	-	-	Nein	Nein
$regexMatch	-	-	Nein	Nein
$replaceOne	-	-	Nein	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
$replaceAll	-	-	Nein	Nein

Systemvariablen

Befehl	3.6	4,0	5.0	Elastic Cluster
$$CURRENT	Nein	Nein	Nein	Nein
$$DESCEND	Ja	Ja	Ja	Ja
$$KEEP	Ja	Ja	Ja	Ja
$$PRUNE	Ja	Ja	Ja	Ja
$$REMOVE	Nein	Nein	Nein	Nein
$$ROOT	Ja	Ja	Ja	Ja

Textsuche-Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$search	Nein	Nein	Ja	Nein
$meta	Nein	Nein	Ja	Nein

Typkonvertierungsoperatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$Konvertieren	Nein	Ja	Ja	Ja
$toBool	Nein	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastic Cluster
$toDate	Nein	Ja	Ja	Ja
$toDecimal	Nein	Ja	Ja	Ja
$toDouble	Nein	Ja	Ja	Ja
$toInt	Nein	Ja	Ja	Ja
$toLong	Nein	Ja	Ja	Ja
$toObjectId	Nein	Ja	Ja	Ja
$toString	Nein	Ja	Ja	Ja
$isNumber	-	-	Nein	Nein

Variablen Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$map	Ja	Ja	Ja	Ja
$let	Ja	Ja	Ja	Ja

Verschiedene Operatoren

Befehl	3.6	4,0	5.0	Elastic Cluster
$rand	-	-	Nein	Nein
$sampleRate	-	-	Nein	Nein
$getField	-	-	Nein	Nein

Datentypen

Befehl	3.6	4,0	5.0	Elastic Cluster
Double	Ja	Ja	Ja	Ja
String	Ja	Ja	Ja	Ja
Object	Ja	Ja	Ja	Ja
Array	Ja	Ja	Ja	Ja
Binäre Daten	Ja	Ja	Ja	Ja
ObjectId	Ja	Ja	Ja	Ja
Boolesch	Ja	Ja	Ja	Ja
Datum	Ja	Ja	Ja	Ja
Null	Ja	Ja	Ja	Ja
32-Bit-Ganzzahl (int)	Ja	Ja	Ja	Ja
Zeitstempel	Ja	Ja	Ja	Ja
64-Bit-Ganzzahl (lang)	Ja	Ja	Ja	Ja
MinKey	Ja	Ja	Ja	Ja
MaxKey	Ja	Ja	Ja	Ja
Decimal128	Ja	Ja	Ja	Ja
Regulärer Ausdruck	Ja	Ja	Ja	Ja
JavaScript	Nein	Nein	Nein	Nein

Befehl	3.6	4,0	5.0	Elastic Cluster
JavaScript(mit Umfang)	Nein	Nein	Nein	Nein
Undefined	Nein	Nein	Nein	Nein
Symbol	Nein	Nein	Nein	Nein
DBPointer	Nein	Nein	Nein	Nein

Indizes und Indexeigenschaften

Themen

Indizes

Befehl	3.6	4,0	5.0	Elastic Cluster
Einzelfeldindex	Ja	Ja	Ja	Ja
Verbundindex	Ja	Ja	Ja	Ja
Multikey-Index	Ja	Ja	Ja	Ja
Textindex	Nein	Nein	Ja	Nein
2dsphere	Ja	Ja	Ja	Nein
2D-Index	Nein	Nein	Nein	Nein
Hash-Index	Nein	Nein	Nein	Nein

Indexeigenschaften

Befehl	3.6	4,0	5.0	Elastic Cluster
TTL	Ja	Ja	Ja	Ja
Unique	Ja	Ja	Ja	Ja
Teilweise	Nein	Nein	Ja	Nein
Berücksichtigt Groß- und Kleinschreibung nicht	Nein	Nein	Nein	Nein
Sparse	Ja	Ja	Ja	Ja
Hintergrund	Ja	Ja	Ja	Nein

Generative künstliche Intelligenz in Amazon DocumentDB

Amazon DocumentDB bietet Funktionen, mit denen Modelle für Machine Learning (ML) und generative künstliche Intelligenz (KI) in Echtzeit mit in Amazon DocumentDB gespeicherten Daten arbeiten können. Kunden müssen keine Zeit mehr mit der Verwaltung einer separaten Infrastruktur, dem Schreiben von Code für die Verbindung mit einem anderen -Service und dem Duplizieren von Daten aus ihrer Primärdatenbank verbringen.

Weitere Informationen zu künstlicher Intelligenz und dazu, wie Ihre KI-Anforderungen unterstützen AWS kann, finden Sie in diesem Artikel „Was ist“. https://aws.amazon.com/what-is/artificial-intelligence/

Themen

- Machine Learning ohne Code mit Amazon SageMaker Canvas
- Vektorsuche nach Amazon DocumentDB

Machine Learning ohne Code mit Amazon SageMaker Canvas

Mit Amazon SageMaker Canvas können Sie Ihre eigenen KI/ML-Modelle erstellen, ohne eine einzige Codezeile schreiben zu müssen. Sie können ML-Modelle für häufige Anwendungsfälle wie Regression und Prognosen erstellen und von Amazon Bedrock aus auf Grundlagenmodelle (FMs) zugreifen und diese aus bewerten. Sie können auch von Amazon aus auf öffentliche FMs SageMaker JumpStart zugreifen, um Inhalte zu generieren, Text zu extrahieren und Text zusammenzufassen, um generative KI-Lösungen zu unterstützen.

So erstellen Sie ML-Modelle ohne Code mit SageMaker Canvas

Amazon DocumentDB ist jetzt in Amazon SageMaker Canvas integriert, um Machine Learning (ML) ohne Code mit Daten zu ermöglichen, die in Amazon DocumentDB gespeichert sind. Sie können jetzt ML-Modelle für Regressions- und Prognoseanforderungen erstellen und Grundlagenmodelle für die Inhaltszusammenfassung und -generierung mithilfe von in Amazon DocumentDB gespeicherten Daten verwenden, ohne eine einzige Codezeile schreiben zu müssen.

SageMaker Canvas bietet eine visuelle Oberfläche, die es Amazon DocumentDB-Kunden ermöglicht, Vorhersagen zu generieren, ohne dass KI/ML-Erfahrung erforderlich ist oder eine einzige Codezeile geschrieben werden muss. Kunden können jetzt den SageMaker Canvas-Workspace über die

starten AWS Management Console, Amazon DocumentDB-Daten für die Datenvorbereitung und das Modelltraining importieren und beitreten. Daten in Amazon DocumentDB können jetzt in SageMaker Canvas verwendet werden, um Modelle zu erstellen und zu erweitern, um Kundenabwanderung vorherzusagen, Betrug zu erkennen, Wartungsfehler vorherzusagen, Geschäftsmetriken zu prognostizieren und Inhalte zu generieren. Kunden können jetzt ML-gesteuerte Erkenntnisse veröffentlichen und teamübergreifend teilen, indem sie die native Integration von SageMaker Canvas mit Amazon verwenden QuickSight. Datenaufnahme-Pipelines in SageMaker Canvas werden standardmäßig auf sekundären Amazon DocumentDB-Instances ausgeführt, wodurch sichergestellt wird, dass die Leistung von Anwendungs- und SageMaker Canvas-Aufnahme-Workloads nicht beeinträchtigt wird.

Amazon-DocumentDB-Kunden können mit SageMaker Canvas beginnen, indem sie zur neuen Seite der Amazon DocumentDB-No-Code-ML-Konsole navigieren und eine Verbindung zu neuen oder verfügbaren SageMaker Canvas-Workspaces herstellen.

Konfigurieren der SageMaker Domain und des Benutzerprofils

Sie können eine Verbindung zu Amazon DocumentDB-Clustern von SageMaker Domänen herstellen, die im Nur-VPC-Modus ausgeführt werden. Durch das Starten einer SageMaker Domain in Ihrer VPC können Sie den Datenfluss aus Ihren SageMaker Studio- und Canvas-Umgebungen steuern. Auf diese Weise können Sie den Internetzugang einschränken, den Datenverkehr mithilfe von AWS Standardnetzwerk- und Sicherheitsfunktionen überwachen und überprüfen und über VPC-Endpunkte eine Verbindung zu anderen - AWS Ressourcen herstellen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon SageMaker Canvas und Konfigurieren von Amazon SageMaker Canvas in einer VPC ohne Internetzugang im Amazon- SageMaker Entwicklerhandbuch, um Ihre SageMaker Domain für die Verbindung mit Ihrem Amazon DocumentDB-Cluster zu erstellen.

Konfigurieren von IAM-Zugriffsberechtigungen für Amazon DocumentDB und SageMaker Canvas

Ein Amazon DocumentDB-Benutzer, der an seine zugeordnete Rolle und Identität `AmazonDocDBConsoleFullAccess` angefügt hat, kann auf die zugreifen AWS Management Console. Fügen Sie der oben genannten Rolle oder Identität die folgenden Aktionen hinzu, um Zugriff auf Machine Learning ohne Code mit Amazon SageMaker Canvas zu gewähren.

```
"sagemaker:CreatePresignedDomainUrl",
"sagemaker:DescribeDomain",
"sagemaker:ListDomains",
```

```
"sagemaker:ListUserProfiles"
```

Erstellen von Datenbankbenutzern und -rollen für SageMaker Canvas

Sie können den Zugriff auf die Aktionen einschränken, die Benutzer mithilfe der rollenbasierten Zugriffskontrolle (RBAC) in Amazon DocumentDB für Datenbanken ausführen können. Bei RBAC werden einem Benutzer eine oder mehrere Rollen gewährt. Diese Rollen bestimmen die Operationen, die ein Benutzer für Datenbankressourcen ausführen kann.

Als Canvas-Benutzer stellen Sie eine Verbindung zu einer Amazon DocumentDB-Datenbank mit Benutzernamen und Passwortanmeldeinformationen her. Sie können einen Datenbankbenutzer/eine Datenbankrolle für einen Canvas-Benutzer erstellen, der/die Lesezugriff auf die spezifischen Datenbanken hat, indem Sie die Amazon-DocumentDBB-RBAC-Funktionalität verwenden.

Verwenden Sie beispielsweise die `-createUser`Operation:

```
db.createUser({
user: "canvas_user",
pwd: "<insert-password>",
roles: [{role: "read", db: "sample-database-1"}]
})
```

Dadurch wird eine erstellt`canvas_user`, die über Leseberechtigungen für die `sample-database-1` Datenbank verfügt. Ihre Canvas-Analysten können diese Anmeldeinformationen verwenden, um auf Daten in Ihrem Amazon DocumentDB-Cluster zuzugreifen. Weitere Datenbankzugriff mit rollenbasierter Zugriffskontrolle Informationen finden Sie unter .

Verfügbare Regionen

Die No-Code-Integration ist in Regionen verfügbar, in denen sowohl Amazon DocumentDB als auch Amazon SageMaker Canvas unterstützt werden. Zu den Regionen gehören:

- us-east-1 (N. Virginia)
- us-east-2 (Ohio)
- us-west-2 (Oregon)
- ap-northeast-1 (Tokyo)
- ap-northeast-2 (Seoul)

- ap-south-1 (Mumbai)
- ap-southeast-1 (Singapur)
- ap-southeast-2 (Sydney)
- eu-central-1 (Frankfurt)
- eu-west-1 (Ireland)

Die neueste Regionsverfügbarkeit finden Sie unter Amazon SageMaker Canvas im Amazon-SageMaker Entwicklerhandbuch.

Vektorsuche nach Amazon DocumentDB

Die Vektorsuche ist eine Methode, die beim Machine Learning verwendet wird, um ähnliche Datenpunkte zu einem bestimmten Datenpunkt zu finden, indem ihre Vektordarstellungen mithilfe von Abstands- oder Ähnlichkeitsmetriken verglichen werden. Je näher sich die beiden Vektoren im Vektorraum befinden, desto ähnlicher gelten die zugrunde liegenden Elemente als. Diese Technik hilft dabei, die semantische Bedeutung der Daten zu erfassen. Dieser Ansatz ist in verschiedenen Anwendungen nützlich, z. B. Empfehlungssysteme, natürliche Sprachverarbeitung und Bilderkennung.

Die Vektorsuche nach Amazon DocumentDB kombiniert die Flexibilität und die umfassende Abfragefunktion einer JSON-basierten Dokumentdatenbank mit der Leistung der Vektorsuche. Wenn Sie Ihre vorhandenen Amazon DocumentDB-Daten oder eine flexible Dokumentdatenstruktur verwenden möchten, um Anwendungsfälle für Machine Learning und generative KI zu erstellen, z. B. semantische Sucherfahrung, Produktempfehlung, Personalisierung, Chatbots, Betrugserkennung und Anomalieerkennung, ist die Vektorsuche nach Amazon DocumentDB eine ideale Wahl für Sie. Die Vektorsuche ist auf Instance-basierten Clustern von Amazon DocumentDB 5.0 verfügbar.

Themen

Einfügen von Vektoren

Um Vektoren in Ihre Amazon DocumentDB-Datenbank einzufügen, können Sie vorhandene Einfügemethoden verwenden:

Beispiel

Im folgenden Beispiel wird eine Sammlung von fünf Dokumenten innerhalb einer Testdatenbank erstellt. Jedes Dokument enthält zwei Felder: den Produktnamen und die entsprechende Vektoreinbettung.

```
db.collection.insertMany([
  {"product_name": "Product A", "vectorEmbedding": [0.2, 0.5, 0.8]},
  {"product_name": "Product B", "vectorEmbedding": [0.7, 0.3, 0.9]},
  {"product_name": "Product C", "vectorEmbedding": [0.1, 0.2, 0.5]},
  {"product_name": "Product D", "vectorEmbedding": [0.9, 0.6, 0.4]},
  {"product_name": "Product E", "vectorEmbedding": [0.4, 0.7, 0.2]}
]);
```

Erstellen eines Vektorindex

Amazon DocumentDB unterstützt sowohl die Hierarchical Navigable Small World (SW)-Indizierung als auch Inverted File mit Flat Compression (IVFFlat)-Indizierungsmethoden. Ein IVFFlat-Index unterteilt Vektoren in Listen und durchsucht anschließend eine ausgewählte Teilmenge dieser Listen, die dem Abfragevektor am nächsten sind. Andererseits organisiert ein SW-Index die Vektordaten in einem mehrschichtigen Diagramm. Obwohl SW im Vergleich zu IVFFlat langsamere Erstellungszeiten aufweist, bietet es eine bessere Abfrageleistung und ein besseres Erinnerungsniveau. Im Gegensatz zu IVFFlat ist SW ohne Trainingsschritt verbunden, sodass der Index ohne anfängliches Laden von Daten generiert werden kann. Für die meisten Anwendungsfälle empfehlen wir, den Indextyp SW für die Vektorsuche zu verwenden.

Wenn Sie keinen Vektorindex erstellen, führt Amazon DocumentDB eine exakte Suche nach dem nächstgelegenen Nachbarn durch, um eine perfekte Erinnerung zu gewährleisten. In Produktionsszenarien ist die Geschwindigkeit jedoch von entscheidender Bedeutung. Wir empfehlen die Verwendung von Vektorindizes, die möglicherweise einen gewissen Erinnerungswert gegen eine verbesserte Geschwindigkeit tauschen. Beachten Sie, dass das Hinzufügen eines Vektorindex zu unterschiedlichen Abfrageergebnissen führen kann.

Vorlagen

Sie können die folgenden - `createIndex` oder -`runCommand`Vorlagen verwenden, um einen Vektorindex auf einem Vektorfeld zu erstellen:

Using createIndex

Bei bestimmten Treibern wie Mongosh und Java `createIndex` kann die Verwendung der `vectorOptions` Parameter in zu einem Fehler führen. In solchen Fällen empfehlen wir die Verwendung von `runCommand`:

```
db.collection.createIndex(
  { "<vectorField>": "vector" },
  { "name": "<indexName>",
    "vectorOptions": {
      "type": " <hnsw> | <ivfflat> ",
      "dimensions": <number_of_dimensions>,
      "similarity": " <euclidean> | <cosine> | <dotProduct> ",
      "lists": <number_of_lists> [applicable for IVFFlat],
      "m": <max number of connections> [applicable for HNSW],
      "efConstruction": <size of the dynamic list for index build> [applicable for
 HNSW]
    }
  }
);
```

Using runCommand

Bei bestimmten Treibern wie Mongosh und Java `createIndex` kann die Verwendung der `vectorOptions` Parameter in zu einem Fehler führen. In solchen Fällen empfehlen wir die Verwendung von `runCommand`:

```
db.runCommand(
  { "createIndexes": "<collection>",
  "indexes": [{
      key: { "<vectorField>": "vector" },
      vectorOptions: {
          type: " <hnsw> | <ivfflat> ",
          dimensions: <number of dimensions>,
          similarity: " <euclidean> | <cosine> | <dotProduct> ",
          lists: <number_of_lists> [applicable for IVFFlat],
          m: <max number of connections> [applicable for HNSW],
          efConstruction: <size of the dynamic list for index build> [applicable for
 HNSW]
```

```
            },
        name: "myIndex"
        }]
    }
);
```

Parameter	Anforderung	Datentyp	Beschreibung	Wert(e)
name	optional	Zeichenfolge	Gibt den Namen des Index an.	Alphanumerisch
type	optional		Gibt den Indextyp an.	Unterstützt: hnsw oder ivfflat Standard: SW (Engine-Patch ab Version 3.0.4574)
dimensions	Erforderlich	Ganzzahl	Gibt die Anzahl der Dimensionen in den Vektordaten an.	Maximal 2 000 Dimensionen.
similarity	Erforderlich	Zeichenfolge	Gibt die Entfernungsmetrik an, die für die Ähnlichkeitsberechnung verwendet wird.	• **euclidean** • **cosine** • **dotProduct**
lists	erforderlich für IVFFlat	Ganzzahl	Gibt die Anzahl der Cluster an, die der IVFFlat-Index verwendet, um die Vektordaten zu gruppieren. Die empfohlen	Minimum: 1 Maximum: Weitere Informationen finden Sie in den Listen pro Instance-

Parameter	Anforderung	Datentyp	Beschreibung	Wert(e)
			e Einstellung ist die Anzahl der Dokumente /1000 für bis zu 1M Dokumente und `sqrt(# of documents )` für über 1M Dokumente.	Typ-Tabelle in Funktionen und Einschränkungen unten.
`m`	optional	Ganzzahl	Gibt die maximale Anzahl von Verbindungen für einen SW-Index an	Standard: 16 Bereich [2, 100]
`efConstruction`	optional	Ganzzahl	Gibt die Größe der dynamischen Kandidatenliste zum Erstellen des Diagramms für den SW-Index an. `efConstruction` muss größer oder gleich (2 * m) sein	Standard: 64 Bereich [4, 1000]

Es ist wichtig, dass Sie den Wert von Unterparametern wie `lists` für IVFFlat und `m efConstruction` für SW entsprechend festlegen, da dies die Genauigkeit/Wiedererkennung, die Aufbauzeit und die Leistung Ihrer Suche beeinträchtigt. Ein höherer Listenwert erhöht die Geschwindigkeit der Abfrage, da er die Anzahl der Vektoren in jeder Liste reduziert, was zu kleineren Regionen führt. Eine kleinere Regionsgröße kann jedoch zu mehr Rückruffehlern führen,

was zu einer geringeren Genauigkeit führt. Bei SW erhöht die Erhöhung des Werts von `m` und `efConstruction` erhöht die Genauigkeit, erhöht aber auch die Index-Build-Zeit und -Größe. Im Folgenden sind einige Beispiele aufgeführt:

Beispiele

HNSW

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "hnsw",
      "dimensions": 3,
      "similarity": "euclidean",
      "m": 16,
      "efConstruction": 64
    }
  }
);
```

IVFFlat

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "ivfflat",
      "dimensions": 3,
      "similarity": "euclidean",
      "lists":1
    }
  }
)
```

Abrufen einer Indexdefinition

Sie können die Details Ihrer Indizes, einschließlich Vektorindizes, mit dem `getIndexes` Befehl anzeigen:

Beispiel

```
db.collection.getIndexes()
```

Beispielausgabe

```
[
 {
  "v" : 4,
  "key" : {
   "_id" : 1
  },
  "name" : "_id_",
  "ns" : "test.collection"
 },
 {
  "v" : 4,
  "key" : {
   "vectorEmbedding" : "vector"
  },
  "name" : "myIndex",
  "vectorOptions" : {
   "type" : "ivfflat",
   "dimensions" : 3,
   "similarity" : "euclidean",
   "lists" : 1
  },
  "ns" : "test.collection"
 }
]
```

Abfragen von Vektoren

Vektorabfragevorlage

Verwenden Sie die folgende Vorlage, um einen Vektor abzufragen:

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": <query vector>,
        "path": "<vectorField>",
```

```
        "similarity": "<distance metric>",
        "k": <number of results>,
        "probes":<number of probes> [applicable for IVFFlat],
        "efSearch":<size of the dynamic list during search> [applicable for HNSW]
      }
    }
  }
]);
```

Parameter	Anforderung	Typ	Beschreibung	Wert(e)
vectorSearch	Erforderlich	operator	Wird innerhalb des Befehls $search verwendet, um die Vektoren abzufragen.	
vector	Erforderlich	Array	Gibt den Abfragevektor an, der verwendet wird, um ähnliche Vektoren zu finden.	
path	Erforderlich	Zeichenfolge	Definiert den Namen des Vektorfelds.	
k	Erforderlich	Ganzzahl	Gibt die Anzahl der Ergebnisse an, die die Suche zurückgibt.	
similarity	Erforderlich	Zeichenfolge	Gibt die Entfernungsmetrik an, die	• **euclidean** • **cosine**

Parameter	Anforderung	Typ	Beschreibung	Wert(e)
			für die Ähnlichkeitsberechnung verwendet wird.	• **`dotProduct`**
`probes`	optional	Ganzzahl	Die Anzahl der Cluster, die die Vektorsuche untersuchen soll. Ein höherer Wert bietet einen besseren Erinnerungswert auf Kosten der Geschwindigkeit. Es kann auf die Anzahl der Listen für die exakt nächste Nachbarnsuche gesetzt werden (an diesem Punkt verwendet der Planer den Index nicht). Die empfohlene Einstellung zum Starten der Feinabstimmung ist `sqrt(# of lists)`.	Standard: 1

Parameter	Anforderung	Typ	Beschreibung	Wert(e)
`efSearch`	optional	Ganzzahl	Gibt die Größe der dynamischen Kandidatenliste an, die der SW-Index während der Suche verwendet. Ein höherer Wert von `efSearch` bietet einen besseren Erinnerungswert zu Kosten der Geschwindigkeit.	Standard: 40 Bereich [1, 1000]

Es ist wichtig, den Wert von `efSearch` (SW) oder `probes` (IVFlat) zu optimieren, um die gewünschte Leistung und Genauigkeit zu erreichen. Sehen Sie sich die folgenden Beispieloperationen an:

HNSW

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "efSearch": 40
      }
    }
  }
]);
```

IVFFlat

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "probes": 1
      }
    }
  }
]);
```

Beispielausgabe

Die Ausgabe dieser Operation sieht in etwa wie folgt aus:

```
{ "_id" : ObjectId("653d835ff96bee02cad7323c"), "product_name" : "Product A",
 "vectorEmbedding" : [ 0.2, 0.5, 0.8 ] }
{ "_id" : ObjectId("653d835ff96bee02cad7323e"), "product_name" : "Product C",
 "vectorEmbedding" : [ 0.1, 0.2, 0.5 ] }
```

Funktionen und Einschränkungen

Versionskompatibilität

- Die Vektorsuche nach Amazon DocumentDB ist nur auf Instance-basierten Clustern von Amazon DocumentDB 5.0 verfügbar.

Vektoren

- Amazon DocumentDB kann Vektoren mit bis zu 2 000 Dimensionen indizieren. Bis zu 16 000 Dimensionen können jedoch ohne Index gespeichert werden.

Indizes

- Für die IVFFlat-Indexerstellung ist die empfohlene Einstellung für den Listenparameter die Anzahl der Dokumente/1000 für bis zu 1M Dokumente und `sqrt(# of documents)` für über 1M Dokumente. Aufgrund eines Arbeitsspeicherlimits unterstützt Amazon DocumentDB je nach Anzahl der Dimensionen einen bestimmten Maximalwert des Listenparameters. Die folgende Tabelle enthält die maximalen Werte des Listenparameters für Vektoren mit 500, 1000 und 2000 Dimensionen:

Instance-Typ	Listen mit 500 Dimensionen	Listen mit 1000 Dimensionen	Listen mit 2000 Dimensionen
t3.med	372	257	150
r5.l	915	741	511
r5.xl	1 393	1 196	901
r5.2xl	5 460	5.230	4 788
r5.4xl	7 842	7 599	7 138
r5.8xl	11.220	10.974	10 498
r5.12xl	13 774	13 526	13 044
r5.16xl	15.943	15 694	15.208
r5.24xl	19 585	19.335	18 845

- Es `partial` werden keine anderen Indexoptionen wie `compound` `sparse` oder mit Vektorindizes unterstützt.
- Paralleler Indexaufbau wird für den SW-Index nicht unterstützt. Es wird nur für den IVFFlat-Index unterstützt.

Vektorabfrage

- Bei Vektorsuchabfragen ist es wichtig, die Parameter wie `probes` oder für `efSearch` optimale Ergebnisse zu optimieren. Je höher der Wert von `probes` oder der `efSearch` Parameter ist, desto höher ist der Erinnerungswert und die Geschwindigkeit verringert sich. Die empfohlene Einstellung, um mit der Feinabstimmung des Sondenparameters zu beginnen, ist `sqrt(# of lists)`.

Bewährte Methoden

Lernen Sie bewährte Methoden für die Arbeit mit Vektorsuchen in Amazon DocumentDB kennen. Dieser Abschnitt wird fortlaufend aktualisiert, wenn neue bewährte Methoden identifiziert werden.

- Die Indexerstellung invertierte Datei mit Flat Compression (IVFFlat) beinhaltet das Clustering und Organisieren der Datenpunkte basierend auf Ähnlichkeiten. Damit ein Index effektiver ist, empfehlen wir daher, dass Sie zumindest einige Daten laden, bevor Sie den Index erstellen.
- Bei Vektorsuchabfragen ist es wichtig, die Parameter wie `probes` oder zu optimieren, um optimale Ergebnisse `efSearch` zu erzielen. Je höher der Wert des `efSearch` Parameters `probes` oder , desto höher ist der Erinnerungswert und niedriger ist die Geschwindigkeit. Die empfohlene Einstellung zum Starten der Feinabstimmung des `probes` Parameters ist `sqrt(lists)`.

Ressourcen

- Vektor durchsucht den neuen Blogbeitrag
- Beispiel für einen semantischen Suchcode
- Beispiele für Amazon DocumentDB-Vektorsuchcodes

Migration zu Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein vollständig verwalteter Datenbankservice, der mit der MongoDB-API kompatibel ist. Sie können Ihre Daten aus MongoDB-Datenbanken, die On-Premises oder in Amazon Elastic Compute Cloud (Amazon EC2) ausgeführt werden, mithilfe des in diesem Abschnitt beschriebenen Prozesses zu Amazon DocumentDB migrieren.

Themen

Aktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service

> **Important**
>
> Amazon DocumentDB folgt nicht denselben Support-Lebenszyklen wie der end-of-life Zeitplan von MongoDB und MongoDB gilt nicht für Amazon DocumentDB . Es gibt keine aktuellen Pläne für end-of-life für für Amazon DocumentDB 3.6, und Ihre vorhandenen MongoDB-3.6-Treiber, Anwendungen und Tools funktionieren weiterhin mit Amazon DocumentDB .

Sie können Ihren Amazon DocumentDB-Cluster mit minimaler Ausfallzeit auf eine höhere Version aktualisierenAWS DMS. AWS DMS ist ein vollständig verwalteter Service, der die Migration von älteren Amazon DocumentDB-Versionen, relationalen Datenbanken und nicht relationalen Datenbanken zu Ihrem Amazon DocumentDB-Zielcluster vereinfacht.

Themen

Schritt 1: Aktivieren von Änderungsstreams

Um eine Migration mit minimalen Ausfallzeiten durchzuführen, AWS DMS benötigt Zugriff auf die Änderungsstreams des Clusters. Amazon DocumentDB-Änderungsstreams bieten eine zeitlich geordnete Abfolge von Aktualisierungsereignissen, die in den Sammlungen und Datenbanken Ihres Clusters auftreten. Das Lesen aus dem Änderungsstream ermöglicht es AWS DMS , Change Data Capture (CDC) durchzuführen und inkrementelle Updates auf den Amazon DocumentDB-Ziel-Cluster anzuwenden.

Um Änderungsstreams für alle Sammlungen in einer bestimmten Datenbank zu aktivieren, authentifizieren Sie sich mit der mongo-Shell bei Ihrem Amazon DocumentDB-Cluster und führen Sie die folgenden Befehle aus:

```
db.adminCommand({modifyChangeStreams: 1,
    database: "db_name",
    collection: "",
    enable: true});
```

Schritt 2: Ändern der Aufbewahrungsdauer für Änderungsstreams

Als Nächstes ändern Sie den Aufbewahrungszeitraum für Änderungsstreams je nachdem, wie lange Sie Änderungsereignisse im Änderungsstream beibehalten möchten. Wenn Sie beispielsweise erwarten, dass Ihre Amazon DocumentDB-Clustermigration mit 12 Stunden AWS DMS dauert, sollten Sie die Aufbewahrung des Änderungsstreams auf einen Wert größer als 12 Stunden festlegen. Der Standardaufbewahrungszeitraum für Ihren Amazon DocumentDB-Cluster beträgt drei Stunden.
Sie können die Aufbewahrungsdauer des Änderungsdatenstrom-Protokolls für Ihren Amazon DocumentDB-Cluster mit der AWS Management Console oder der auf eine Stunde bis sieben Tage ändernAWS CLI. Weitere Informationen finden Sie unter Ändern der Aufbewahrungsdauer von Stream-Protokollen.

Schritt 3: Migrieren Ihrer Indizes

Erstellen Sie dieselben Indizes auf Ihrem Amazon DocumentDB-Ziel-Cluster wie auf Ihrem Amazon DocumentDB-Quell-Cluster. Obwohl die Migration von Daten AWS DMS übernimmt, werden keine Indizes migriert. Um die Indizes zu migrieren, verwenden Sie das Amazon DocumentDB-Index-Tool, um Indizes aus dem Amazon DocumentDB-Quellcluster zu exportieren. Sie können das Tool abrufen, indem Sie einen Klon des Amazon DocumentDB-Tools GitHub -Repositorys erstellen und die Anweisungen unter befolgen`README.md`. Sie können das Tool von einer Amazon EC2-Instance oder einer AWS Cloud9 Umgebung ausführen, die in derselben Amazon VPC wie Ihr Amazon DocumentDB-Cluster ausgeführt wird.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Der folgende Code-Dumping-Indizes aus Ihrem Amazon DocumentDB-Quell-Cluster:

```
python migrationtools/documentdb_index_tool.py --dump-indexes
--uri mongodb://sample-user:user-password@sample-source-cluster.node.us-
east 1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
 21:46:50,432: Successfully connected to instance docdb-40-xx.cluster-xxxxxxxx.us-
east-1.docdb.amazonaws.com:27017
2020-02-11 21:46:50,432: Retrieving indexes from server...2020-02-11 21:46:50,440:
 Completed writing index metadata to local folder: /home/ec2-user/index.js/
```

Sobald Ihre Indizes erfolgreich exportiert wurden, stellen Sie diese Indizes in Ihrem Amazon DocumentDB-Ziel-Cluster wieder her. Um die Indizes wiederherzustellen, die Sie im vorherigen Schritt exportiert haben, verwenden Sie das Amazon DocumentDB Index Tool. Der folgende Befehl stellt die Indizes in Ihrem Amazon DocumentDB-Ziel-Cluster aus dem angegebenen Verzeichnis wieder her.

```
python migrationtools/documentdb_index_tool.py --restore-indexes
--uri mongodb://sample-user:user-password@sample-destination-
cluster.node.us-east 1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
 21:51:23,245: Successfully connected to instance docdb-50-xx.cluster-xxxxxxxx.us-
east-1.docdb.amazonaws.com:27017
2020-02-11 21:51:23,264: testdb.coll: added index: _id
```

Um zu bestätigen, dass Sie die Indizes korrekt wiederhergestellt haben, stellen Sie eine Verbindung zu Ihrem Amazon DocumentDB-Ziel-Cluster mit der mongo-Shell her und listen Sie die Indizes für eine bestimmte Sammlung auf. Sehen Sie sich den folgenden Code an:

```
mongo --ssl
--host docdb-xx-xx.cluster-xxxxxxxx.us-east-1.docdb.amazonaws.com:27017
--sslCAFile rds-ca-2019-root.pem --username documentdb --password documentdb

db.coll.getIndexes()
```

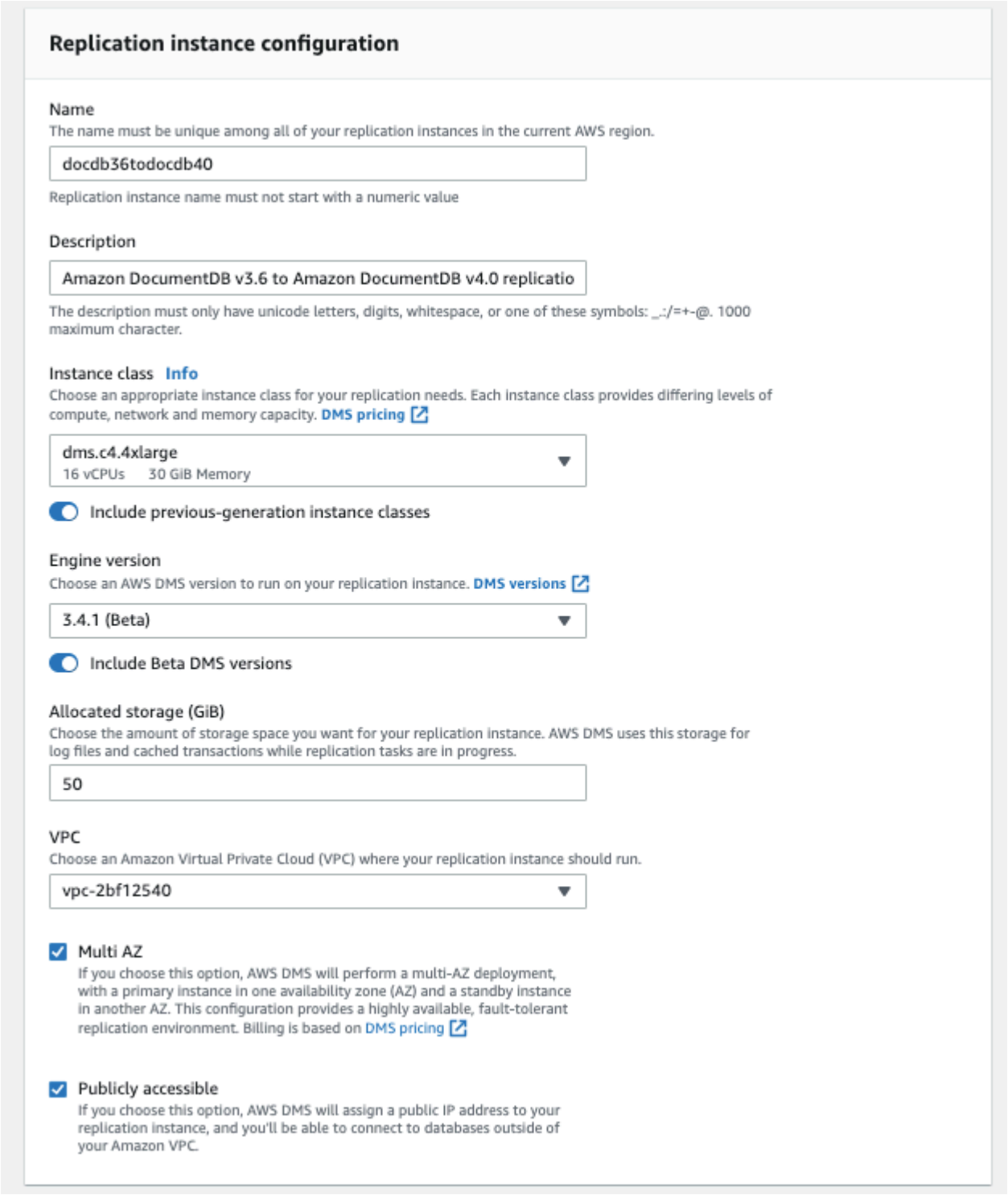
Schritt 4: Erstellen einer AWS DMS Replikations-Instance

Eine AWS DMS Replikations-Instance verbindet und liest Daten aus Ihrem Amazon DocumentDB-Quell-Cluster und schreibt sie in Ihren Amazon DocumentDB-Ziel-Cluster. Die AWS DMS Replikations-Instance kann sowohl Massenladevorgänge als auch CDC-Operationen ausführen. Der Großteil dieser Verarbeitung findet im Speicher statt. Große Operationen erfordern jedoch möglicherweise eine gewisse Pufferung auf der Festplatte. Zwischengespeicherte Transaktionen und Protokolldateien werden ebenfalls auf Festplatte geschrieben. Sobald die Daten migriert wurden, streamt die Replikations-Instance auch alle Änderungsereignisse, um sicherzustellen, dass die Quelle und das Ziel synchron sind.

So erstellen Sie eine AWS DMS Replikations-Instance:

1. Öffnen Sie die AWS DMS-Konsole.
2. Wählen Sie im Navigationsbereich Replication instances (Replikations-Instances) aus.
3. Wählen Sie Create replication instance (Replikations-Instance erstellen) aus und geben Sie die folgenden Informationen ein:

   - Geben Sie unter Name einen Namen Ihrer Wahl ein. Beispiel: `docdb36todocdb40`
   - Geben Sie unter Beschreibung eine Beschreibung Ihrer Wahl ein. Für listitem, Amazon DocumentDB 3.6 zu Amazon DocumentDB 4.0 Replikations-Instance.
   - Wählen Sie für Instance-Klasse die Größe aus, die Ihren Anforderungen entspricht.
   - Wählen Sie für Engine-Version die Option `3.4.1.`
   - Wählen Sie für Amazon VPC die Amazon VPC aus, die Ihre Quell- und Ziel-Amazon DocumentDB-Cluster enthält.
   - Verwenden Sie für Zugewiesener Speicher (GiB) den Standardwert von 50 GiB . Wenn Sie einen hohen Schreibdurchsatz-Workload haben, erhöhen Sie diesen Wert entsprechend Ihrem Workload.
   - Wählen Sie für Multi-AZ die Option Ja aus, wenn Sie Hochverfügbarkeit und Failover-Unterstützung benötigen.
   - Für Publicly accessible (Öffentlich zugänglich) aktivieren Sie diese Option.



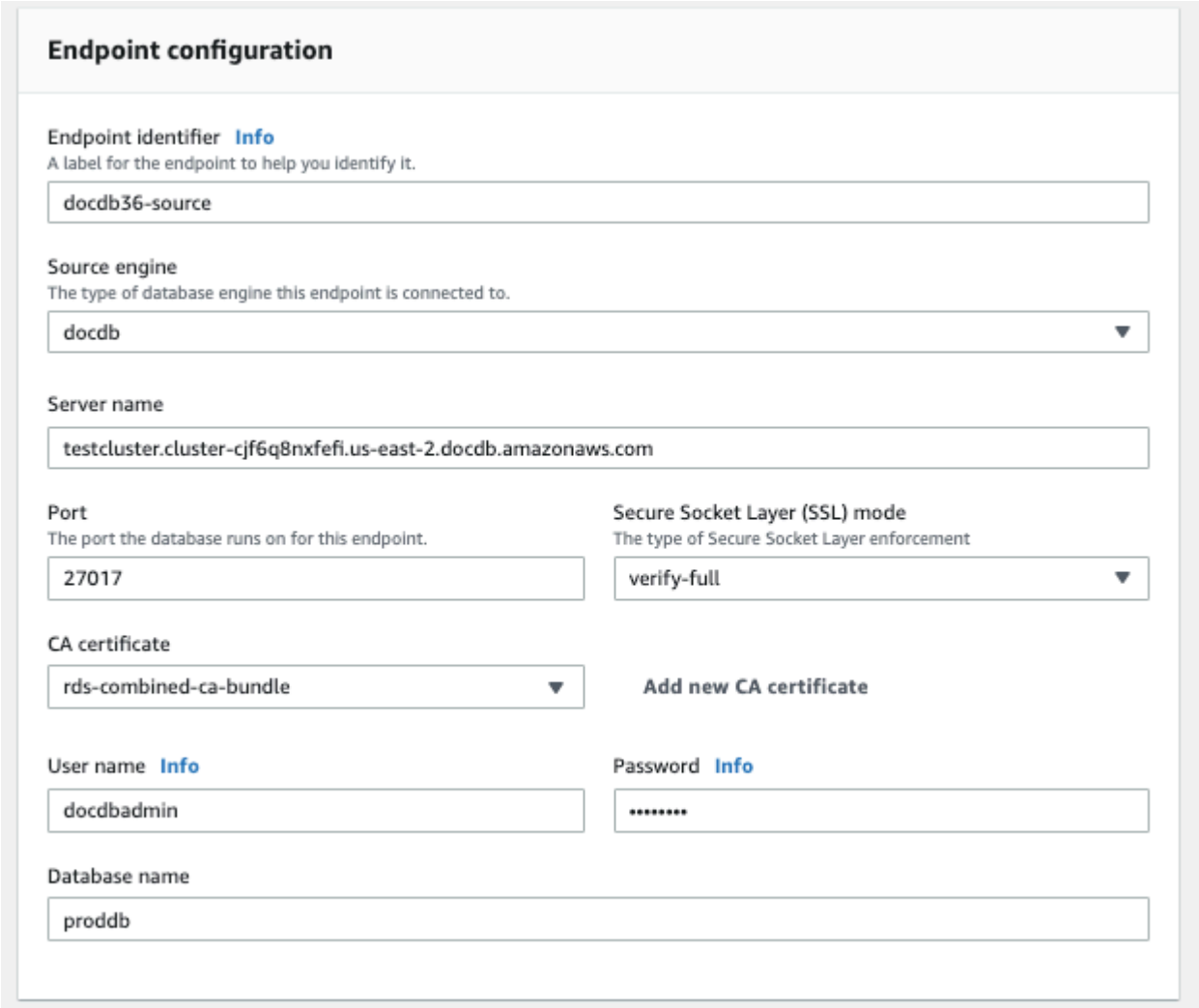
4. Wählen Sie Create replication instance (Replikations-Instance erstellen) aus.

Schritt 5: Erstellen eines -AWS DMSQuellendpunkts

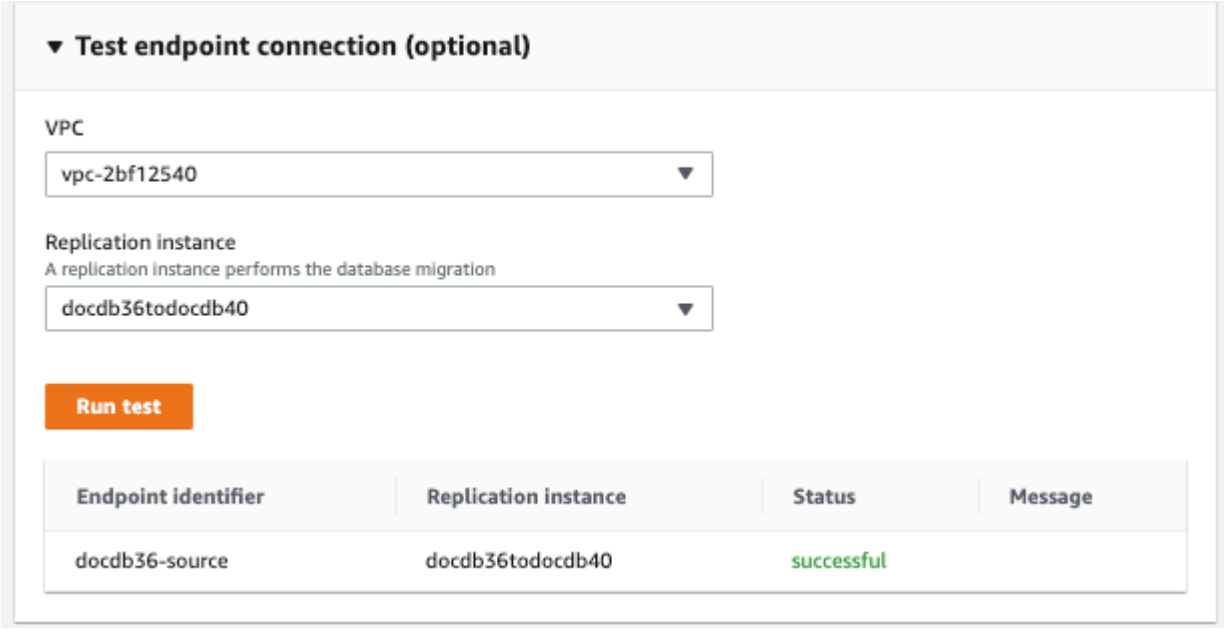
Der Quellendpunkt wird für den Amazon DocumentDB-Quell-Cluster verwendet.

So erstellen Sie einen Quellendpunkt

1. Öffnen Sie die AWS DMS-Konsole.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie die folgenden Informationen aus `Create endpoint` und geben Sie sie ein:
   - Für Endpoint type (Endpunkttyp) wählen Sie Source (Quelle) aus.
   - > Geben Sie für Endpunkt-ID einen Namen ein, den Sie sich leicht merken können, z. B. `docdb-source`.
   - Wählen Sie für Quell-Engine die Option aus`docdb`.
   - Geben Sie für Servername den DNS-Namen Ihres Amazon DocumentDB-Quell-Clusters ein.
   - Geben Sie für Port die Portnummer Ihres Amazon DocumentDB-Quell-Clusters ein.
   - Wählen Sie für SSL-Modus `verify-full`.
   - Wählen Sie für CA-Zertifikat die Option Neues CA-Zertifikat hinzufügen aus. Laden Sie das neue CA-Zertifikat , um das TLS-Verbindungspaket zu erstellen. Geben Sie für Zertifikat-ID `rds-combined-ca-bundle` ein. Wählen Sie unter Import certificate file (Zertifikatsdatei importieren) die Option Choose file (Datei auswählen) und navigieren Sie zu der `.pem`-Datei, die Sie zuvor heruntergeladen haben. Wählen Sie die Datei aus und öffnen Sie sie. Wählen Sie Zertifikat importieren und dann `rds-combined-ca-bundle` aus der Dropdownliste Zertifikat auswählen aus.
   - Geben Sie für Benutzername den Hauptbenutzernamen Ihres Amazon DocumentDB-Quell-Clusters ein.
   - Geben Sie für Passwort das Masterpasswort Ihres Amazon DocumentDB-Quell-Clusters ein.
   - Geben Sie für Datenbankname den Datenbanknamen ein, den Sie aktualisieren möchten.



4. Testen Sie Ihre Verbindung, um sicherzustellen, dass sie erfolgreich eingerichtet wurde.



5. Klicken Sie auf Endpunkt erstellen.

> **Note**
>
> AWS DMS kann jeweils nur eine Datenbank migrieren.

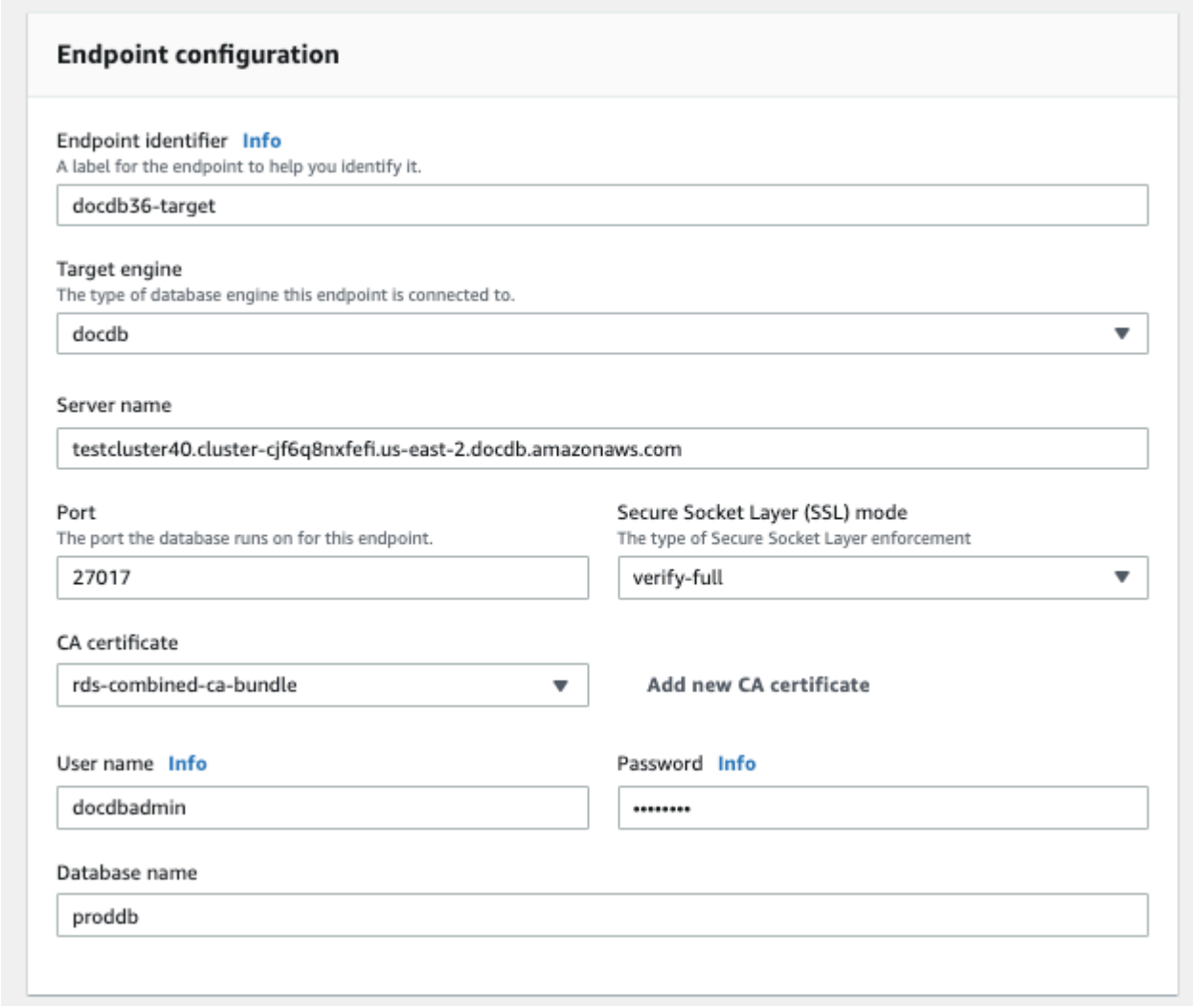
Schritt 6: Erstellen eines AWS DMS Zielendpunkts

Der Zielendpunkt ist für Ihren Amazon DocumentDB-Ziel-Cluster bestimmt.

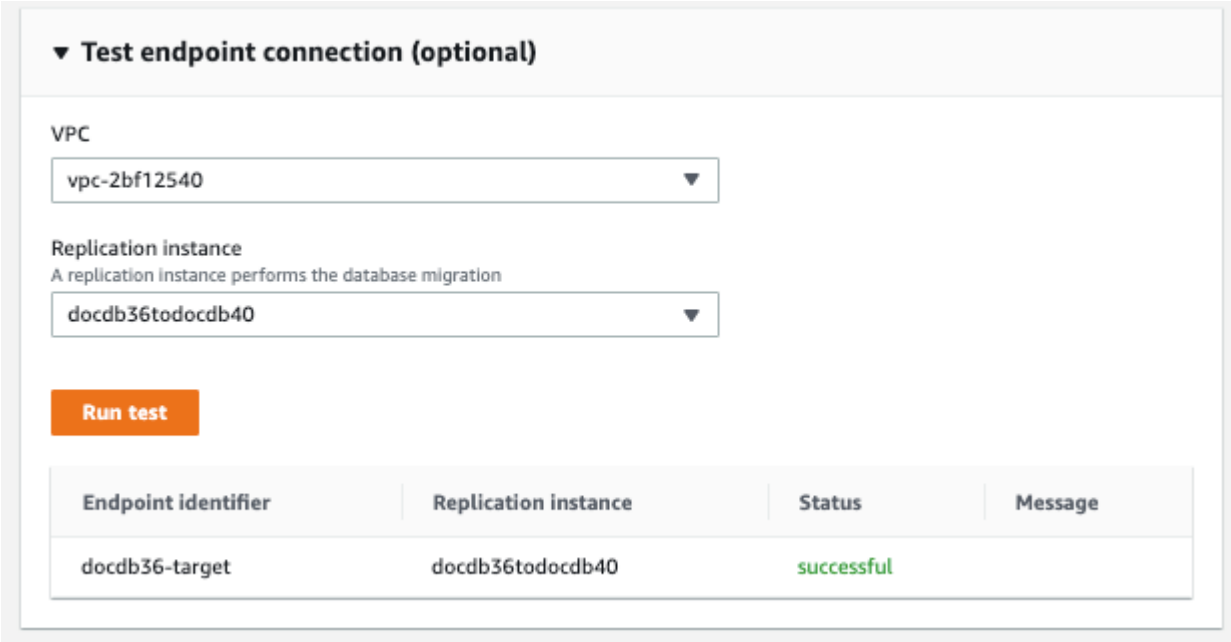
So erstellen Sie einen Zielendpunkt:

1. Öffnen Sie die AWS DMS-Konsole.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Create endpoint (Endpunkt erstellen) und geben Sie die folgenden Informationen ein:
   - Für Endpoint type (Endpunkttyp) wählen Sie Target (Ziel) aus.
   - Für Endpoint Identifier (Endpunkt-ID) geben Sie einen Namen ein, der leicht zu merken ist, zum Beispiel `docdb-target`.
   - Wählen Sie für Quell-Engine die Option aus`docdb`.
   - Geben Sie für Servername den DNS-Namen Ihres Amazon DocumentDB-Ziel-Clusters ein.

- Geben Sie für Port die Portnummer Ihres Amazon DocumentDB-Ziel-Clusters ein.
- Wählen Sie für SSL-Modus `verify-full`.
- Wählen Sie für CA-Zertifikat das vorhandene `rds-combined-ca-bundle` Zertifikat aus der Dropdownliste Zertifikat auswählen aus.
- Geben Sie für Benutzername den Hauptbenutzernamen Ihres Amazon DocumentDB-Ziel-Clusters ein.
- Geben Sie für Passwort das Hauptpasswort Ihres Amazon DocumentDB-Ziel-Clusters ein.
- Geben Sie für Datenbankname denselben Datenbanknamen ein, den Sie für die Einrichtung Ihres Quellendpunkts verwendet haben.



4. Testen Sie Ihre Verbindung, um sicherzustellen, dass sie erfolgreich eingerichtet wurde.



5. Klicken Sie auf Endpunkt erstellen.

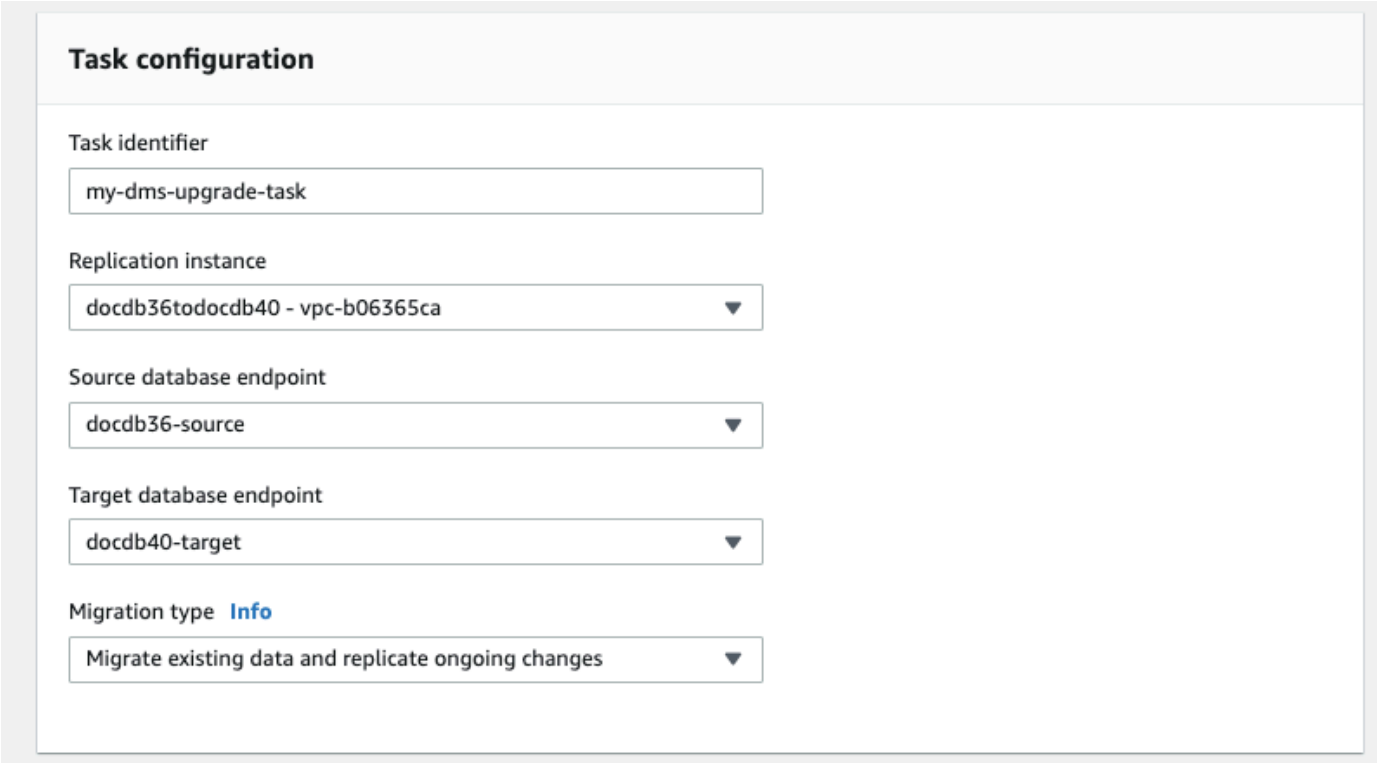
Schritt 7: Erstellen und Ausführen einer Migrationsaufgabe

Eine -AWS DMSAufgabe bindet die Replikations-Instance an Ihre Quell- und Ziel-Instance. Wenn Sie eine Migrationsaufgabe erstellen, geben Sie den Quellendpunkt, den Zielendpunkt, die Replikations-Instance und alle gewünschten Migrationseinstellungen an. Eine -AWS DMSAufgabe kann mit drei verschiedenen Migrationstypen erstellt werden: Migrieren vorhandener Daten, Migrieren vorhandener Daten und Replizieren laufender Änderungen oder nur Replizieren von Datenänderungen. Da der Zweck dieser Anleitung darin besteht, einen Amazon DocumentDB-Cluster mit minimaler Ausfallzeit zu aktualisieren, verwenden die Schritte die Option , vorhandene Daten zu migrieren und laufende Änderungen zu replizieren. Mit dieser Option erfasst AWS DMS Änderungen während der Migration Ihrer vorhandenen Daten. Selbst nach dem Laden der Massendaten erfasst AWS DMS weiterhin Änderungen und wendet diese an. Schließlich sind die Quell- und Zieldatenbanken synchronisiert, unter Berücksichtigung einer minimalen Ausfallzeit während der Migration.

Im Folgenden finden Sie die Schritte zum Erstellen einer Migrationsaufgabe für eine Migration mit minimalen Ausfallzeiten:

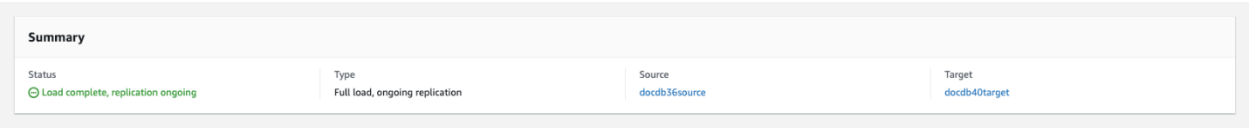
1. Öffnen Sie die AWS DMS-Konsole.
2. Wählen Sie im Navigationsbereich Tasks aus.
3. Wählen Sie Create task (Aufgabe erstellen) und geben Sie die folgenden Informationen ein:

- Für Task name (Aufgabenname) geben Sie einen Namen ein, der leicht zu merken ist, zum Beispiel `my-dms-upgrade-task`.
- Wählen Sie für Replikations-Instance die Replikations-Instance aus, die Sie in Step3: Erstellen einer AWS Database Migration Service Replikations-Instance erstellt haben.
- Wählen Sie für Quellendpunkt den Quellendpunkt aus, den Sie in Schritt 4: Erstellen eines AWS Database Migration Service Quellendpunkts erstellt haben
- Wählen Sie für Zielendpunkt den Zielendpunkt aus, den Sie in Schritt 5: Erstellen eines AWS Database Migration Service Zielendpunkts erstellt haben
- Wählen Sie für Migrationstyp die Option Vorhandene Daten migrieren und laufende Änderungen replizieren aus.



4. Aktivieren Sie im Abschnitt Aufgabeneinstellungen CloudWatch die Protokolle .
5. Wählen Sie im Abschnitt Tabellenzuordnungen die Option Nichts tun aus. Dadurch wird sichergestellt, dass die in Schritt 3 erstellten Indizes nicht gelöscht werden.
6. Wählen Sie für die Startkonfiguration der Migrationsaufgabe Automatisch beim Erstellen aus. Dadurch wird die Migrationsaufgabe automatisch gestartet, sobald Sie sie erstellt haben.
7. Wählen Sie Create task aus.

AWS DMS beginnt jetzt mit der Migration von Daten von Ihrem Amazon DocumentDB-Quellcluster zu Ihrem Amazon DocumentDB-Zielcluster. Der Aufgabenstatus sollte sich von Wird gestartet in Wird ausgeführt ändern. Sie können den Fortschritt überwachen, indem Sie Aufgaben in der AWS DMS Konsole auswählen. Nach einigen Minuten/Stunden (abhängig von der Größe Ihrer Migration) sollte sich der Status von in Laden abgeschlossen ändern. Die Replikation läuft weiter. Das bedeutet, dass eine Volllastmigration Ihres Amazon DocumentDB-Quell-Clusters zu einem Amazon DocumentDB-Ziel-Cluster abgeschlossen AWS DMS hat und jetzt Änderungsereignisse repliziert.



Schließlich werden Quelle und Ziel synchronisiert. Sie können überprüfen, ob sie synchron sind, indem Sie eine `-count()`Operation für Ihre Sammlungen ausführen, um zu überprüfen, ob alle Änderungsereignisse migriert wurden.

Schritt 8: Ändern des Anwendungsendpunkts in den Amazon DocumentDB-Ziel-Cluster

Nachdem der vollständige Ladevorgang abgeschlossen ist und der CDC-Prozess kontinuierlich repliziert wird, können Sie den Datenbankverbindungsendpunkt Ihrer Anwendung von Ihrem Amazon DocumentDB-Quellcluster in Ihren Amazon DocumentDB-Zielcluster ändern.

Migrationstools

Um zu Amazon DocumentDB zu migrieren, verwenden die meisten Kunden die AWS Database Migration Service (AWS DMS) und Befehlszeilen-Dienstprogramme wie `mongodump` und `mongorestore`. Als bewährte Methode und für eine dieser Optionen empfehlen wir Ihnen, zuerst Indizes in Amazon DocumentDB zu erstellen, bevor Sie mit Ihrer Migration beginnen, da dies die Gesamtzeit reduzieren und die Geschwindigkeit der Migration erhöhen kann. Dazu können Sie das Amazon DocumentDB Index Tool verwenden.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) ist ein Cloud-Service, der die Migration relationaler Datenbanken und nicht relationaler Datenbanken zu Amazon DocumentDB vereinfacht. Sie können verwendenAWS DMS, um Ihre Daten aus Datenbanken, die On-Premises oder auf EC2 gehostet werden, zu Amazon DocumentDB zu migrieren. Mit können AWS DMSSie einmalige Migrationen durchführen oder laufende Änderungen replizieren, um Quellen und Ziele synchron zu halten.

Weitere Informationen zur Verwendung von AWS DMS für die Migration zu Amazon DocumentDB finden Sie unter:

- Verwenden von MongoDB als Quelle für AWS DMS
- Verwenden von Amazon DocumentDB als Ziel für AWS Database Migration Service
- Walkthrough: Migrieren von MongoDB zu Amazon DocumentDB

Befehlszeilen-Dienstprogramme

Zu den gängigen Dienstprogrammen für die Migration von Daten zu und von Amazon DocumentDB gehören `mongodump`, `mongorestoremongoexport`, und `mongoimport`. In der Regel sind `mongodump` und `mongorestore` die effizientesten Dienstprogramme, da sie Daten aus Ihren Datenbanken in einem binären Format sichern und wiederherstellen. Dies ist im Allgemeinen die leistungsstärkste Option und ergibt eine geringere Datenmenge im Vergleich zu logischen Exporten. `mongoexport` und `mongoimport` sind nützlich, wenn Sie Daten in einem logischen Format wie JSON oder CSV exportieren und importieren möchten, da die Daten menschlich lesbar sind. Diese Option ist im Allgemeinen jedoch langsamer als `mongodump`/`mongorestore` und ergibt eine größere Datenmenge.

Im folgenden Migrationsansätze Abschnitt wird beschrieben, wann Sie - AWS DMS und - Befehlszeilen-Dienstprogramme basierend auf Ihrem Anwendungsfall und Ihren Anforderungen am besten verwenden können.

Erkennung

Für jede Ihrer MongoDB-Bereitstellungen sollten Sie zwei Datasets identifizieren und protokollieren: Architekturdetails und Betriebseigenschaften. Diese Informationen helfen Ihnen bei der Auswahl des geeigneten Migrationsansatzes und der Clustergröße.

Architekturdetails

- Name

  Wählen Sie einen eindeutigen Namen für die Verfolgung dieser Bereitstellung aus.

- Version

Erfassen Sie die Version von MongoDB, die Ihre Bereitstellung ausführt. Um die Version zu finden, verbinden Sie sich über die Mongo-Shell mit einem Mitglied des Replikatsatzes und führen die Operation `db.version()` aus.

- Typ

  Protokollieren Sie, ob es sich bei Ihrer Bereitstellung um eine eigenständige Mongo-Instance, einen Replikatsatz oder einen Sharded-Cluster handelt.

- Mitglieder

  Protokollieren Sie die Hostnamen, Adressen und Ports der einzelnen Cluster, Replikatsätze oder eigenständigen Mitglieder.

  Für eine geclusterte Bereitstellung können Sie Shard-Mitglieder finden, indem Sie sich per Mongo-Shell mit einem Mongo-Host verbinden und die Operation `sh.status()` ausführen.

  Sie können die Mitglieder eines Replikatsatzes abrufen, indem Sie sich per Mongo-Shell mit einem Replikatsatz verbinden und die Operation `rs.status()` ausführen.

- Oplog-Größen

  Bei Replikatsätzen oder Sharded-Clustern notieren Sie die Größe des Oplogs für jedes Replikatsatzmitglied. Um die Oplog-Größe eines Mitglieds zu finden, verbinden Sie sich mit dem Replikat-Set-Element mit der Mongo-Shell und führen Sie die Operation `ps.printReplicationInfo()` aus.

- Prioritäten für Replikatsatzmitglieder

  Protokollieren Sie bei den Replikatsätzen oder Sharded-Clustern die Priorität der einzelnen Replikatsatzmitglieder. Um die Prioritäten für das Replikatsatzmitglied zu finden, verbinden Sie sich

per Mongo-Shell mit einem Replikatsatzmitglied und führen Sie die Operation `rs.conf()` aus. Die Priorität wird als Wert des Schlüssels `priority` angezeigt.

- TLS/SSL-Nutzung

  Protokollieren Sie, ob auf den einzelnen Knoten während der Übertragung TLS/SSL (Transport Layer Security/Secure Sockets Layer) verwendet wird.

Betriebseigenschaften

- Datenbankstatistik

  Protokollieren Sie für jede Sammlung die folgenden Informationen:

  - Name
  - Datengröße
  - Sammlungsanzahl

  Um die Datenbankstatistiken zu finden, verbinden Sie sich per Mongo-Shell mit Ihrer Datenbank und führen Sie den Befehl `db.runCommand({dbstats: 1})` aus.

- Sammlungsstatistik

  Protokollieren Sie für jede Sammlung die folgenden Informationen:

  - Namespace
  - Datengröße
  - Indexanzahl
  - Ob die Sammlung gedeckelt ist

- Indexstatistik

  Erfassen Sie für jede Sammlung die folgenden Indexinformationen:

  - Namespace

- ID
- Größe
- Schlüssel
- TTL
- Sparse
- Hintergrund

Um die Indexinformationen zu finden, verbinden Sie sich per Mongo-Shell mit Ihrer Datenbank und führen Sie den Befehl `db.collection.getIndexes()` aus.

- Opcounters

  Diese Informationen helfen Ihnen, Ihre aktuellen MongoDB-Workload-Muster (viele Lesevorgänge, viele Schreibvorgänge oder ausgeglichen) zu ermitteln. Es bietet auch Anleitungen zu Ihrer anfänglichen Amazon DocumentDB-Instance-Auswahl.

  Nachfolgend sind die wichtigsten Informationen aufgeführt, die während des Überwachungszeitraums (in der Form Anzahl/Sekunde) gesammelt werden müssen:

  - Abfragen
  - Einfügungen
  - Aktualisierungen
  - Löschvorgänge

  Sie können diese Informationen abrufen, indem Sie die Ausgabe des Befehls `db.serverStatus()` über den gewünschten Zeitraum grafisch darstellen. Sie können außerdem das Mongostat-Toolverwenden, um direkt Werte für diese Statistiken zu erhalten. Mit dieser Option laufen Sie jedoch Gefahr, Ihre Migration auf Basis von Nutzungszeiträumen zu planen, die keine Spitzenauslastung darstellen.

- Netzwerkstatistik

  Diese Informationen helfen Ihnen, Ihre aktuellen MongoDB-Workload-Muster (viele Lesevorgänge, viele Schreibvorgänge oder ausgeglichen) zu ermitteln. Es bietet auch Anleitungen zu Ihrer anfänglichen Amazon DocumentDB-Instance-Auswahl.

  Nachfolgend sind die wichtigsten Informationen aufgeführt, die während des Überwachungszeitraums (in der Form Anzahl/Sekunde) gesammelt werden müssen:

  - Verbindungen
  - Netzwerk-Bytes eingehend
  - Netzwerk-Bytes ausgehend

  Sie können diese Informationen abrufen, indem Sie die Ausgabe des Befehls `db.serverStatus()` über den gewünschten Zeitraum grafisch darstellen. Sie können außerdem das Mongostat-Toolverwenden, um direkt Werte für diese Statistiken zu erhalten. Mit dieser Option laufen Sie jedoch Gefahr, Ihre Migration auf Basis von Nutzungszeiträumen zu planen, die keine Spitzenauslastung darstellen.

Planung: Amazon DocumentDB-Cluster-Anforderungen

Bei einer erfolgreichen Migration müssen Sie sowohl die Konfiguration Ihres Amazon DocumentDB-Clusters als auch den Zugriff von Anwendungen auf Ihren Cluster sorgfältig abwägen. Berücksichtigen Sie bei der Ermittlung Ihrer Cluster-Anforderungen die folgenden Dimensionen:

- Verfügbarkeit

  Amazon DocumentDB bietet hohe Verfügbarkeit durch die Bereitstellung von Replikat-Instances, die in einem Prozess, der als Failover bezeichnet wird, zu einer primären Instance hochgestuft werden können. Durch die Bereitstellung von Replikations-Instances in verschiedenen Availability Zones können Sie eine höhere Verfügbarkeit erreichen.

Die folgende Tabelle enthält Richtlinien für Amazon DocumentDB-Bereitstellungskonfigurationen zur Erfüllung bestimmter Verfügbarkeitsziele.

Verfügbarkeitsziel	Gesamtzahl der Instances	Replikas	Availability Zones
99 %	1	0	1
99,9 %	2	1	2
99,99 %	3	2	3

Für die allgemeine Systemzuverlässigkeit müssen alle Komponenten berücksichtigt werden – nicht nur die Datenbank. Bewährte Methoden und Empfehlungen zur Erfüllung der allgemeinen Anforderungen an die Systemzuverlässigkeit finden Sie im AWS Well-Architected Reliability Pillar Whitepaper .

- Leistung

  Amazon DocumentDB-Instances ermöglichen es Ihnen, aus dem Speicher-Volume Ihres Clusters zu lesen und darin zu schreiben. Es gibt unterschiedliche Typen von Cluster-Instances mit verschiedenen Speicher- und vCPU-Konfigurationen, die sich auf die Lese- und Schreibleistung Ihres Clusters auswirken. Wählen Sie anhand der in der Ermittlungsphase gesammelten Informationen einen Instance-Typ aus, der Ihre Anforderungen an die Workload-Performance abdeckt. Eine Liste mit unterstützten Instance-Typen finden Sie unter Verwalten von Instance-Klassen.

  Berücksichtigen Sie bei der Auswahl eines Instance-Typs für Ihren Amazon DocumentDB-Cluster die folgenden Aspekte der Leistungsanforderungen Ihres Workloads:

- vCPUs – Architekturen, die eine höhere Verbindungsanzahl erfordern, können von Instances mit mehr vCPUS profitieren.

- Arbeitsspeicher – Wenn möglich, bietet das Speichern Ihres Arbeitsdatensatzes im Arbeitsspeicher maximale Leistung. Eine erste Richtlinie besteht darin, ein Drittel des Speichers Ihrer Instance für die Amazon DocumentDB-Engine zu reservieren, wobei zwei Drittel für Ihren Arbeitsdatensatz übrig bleiben.

- Verbindungen – Die minimale optimale Verbindungsanzahl beträgt acht Verbindungen pro vCPU der Amazon DocumentDB-Instance. Obwohl das Verbindungslimit für Amazon DocumentDB-Instances viel höher ist, sinken die Leistungsvorteile zusätzlicher Verbindungen über acht Verbindungen pro vCPU.

- Netzwerk – Workloads mit einer großen Anzahl von Clients oder Verbindungen sollten die Gesamtnetzwerkleistung berücksichtigen, die für eingefügte und abgerufene Daten erforderlich ist. Massenoperationen können die Netzwerkressourcen effizienter nutzen.

- Leistung einfügen – Einfügungen einzelner Dokumente sind im Allgemeinen die langsamste Methode, um Daten in Amazon DocumentDB einzufügen. Masseneinfügeoperationen können wesentlich schneller sein.

- Leseleistung – Lesevorgänge aus dem Arbeitsspeicher sind immer schneller als Lesevorgänge, die vom Speichervolume zurückgegeben werden. Daher ist es ideal, die Größe des Instance-Arbeitsspeichers zu optimieren, um das Arbeits-Dataset im Arbeitsspeicher zu halten.

Zusätzlich zur Bereitstellung von Lesevorgängen aus Ihrer primären Instance werden Amazon DocumentDB-Cluster automatisch als Replikatsätze konfiguriert. Sie können dann schreibgeschützte Abfragen an Lesereplikate weiterleiten, indem Sie die Leseeinstellung in Ihrem MongoDB-Treiber festlegen. Sie können den Leseverkehr skalieren, indem Sie Replikate hinzufügen, was die Gesamtbelastung der primären Instance reduziert.

Es ist möglich, Amazon DocumentDB-Replikate verschiedener Instance-Typen im selben Cluster bereitzustellen. Ein exemplarischer Anwendungsfall könnte sein, eine Replik mit einem größeren Instance-Typ zu erstellen, um temporären Analyse-Datenverkehr zu verarbeiten. Wenn Sie einen gemischten Satz von Instance-Typen bereitstellen, sorgen Sie dafür, dass die Failover-Priorität für jede Instance konfiguriert ist. Dadurch wird sichergestellt, dass ein Failover-Ereignis stets ein Replikat ausreichender Größe für Ihre Schreiblast hochstuft.

- Wiederherstellung

  Amazon DocumentDB sichert Ihre Daten kontinuierlich, während sie geschrieben werden. Es bietet point-in-time Wiederherstellungsfunktionen (PITR) innerhalb eines konfigurierbaren Zeitraums von 1 bis 35 Tagen, die als Aufbewahrungszeitraum für Backups bezeichnet werden. Der Standardaufbewahrungszeitraum für Backups beträgt einen Tag. Amazon DocumentDB erstellt außerdem automatisch tägliche Snapshots Ihres Speicher-Volumes, die auch für den konfigurierten Aufbewahrungszeitraum für Backups aufbewahrt werden.

  Wenn Sie Snapshots über den Aufbewahrungszeitraum für Backups hinaus speichern möchten, können Sie mit AWS Management Console und AWS Command Line Interface (AWS CLI) jederzeit manuelle Snapshots initiieren. Weitere Informationen finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.

  Beachten Sie bei der Planung Ihrer Migration Folgendes:

  - Wählen Sie einen Aufbewahrungszeitraum für Backups von 1 bis 35 Tagen, der Ihrem Recovery Point Objective (RPO) entspricht.
  - Entscheiden Sie, ob und in welchem Intervall Sie manuelle Snapshots benötigen.

Migrationsansätze

Es gibt drei primäre Ansätze für die Migration Ihrer Daten zu Amazon DocumentDB .

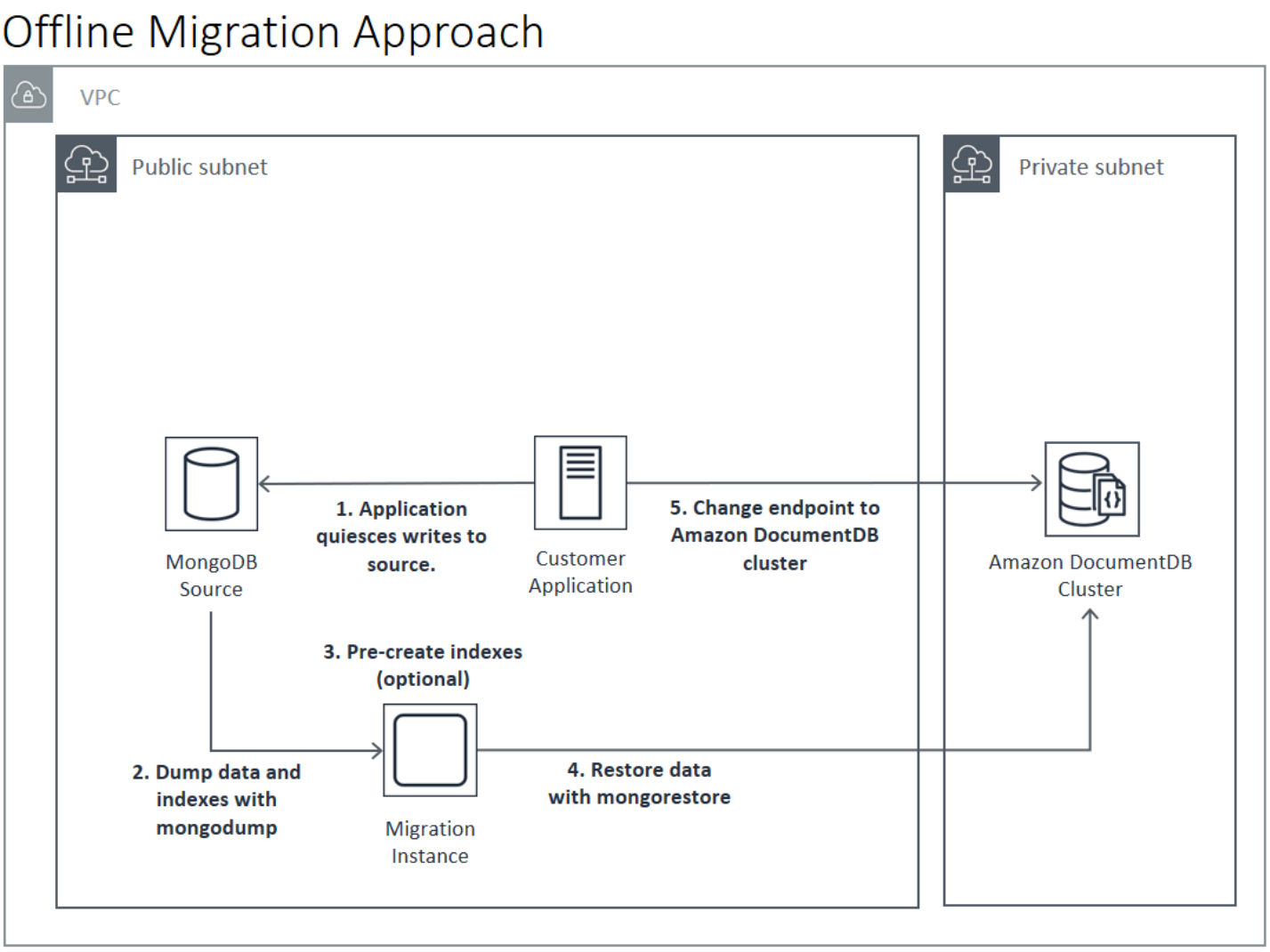
> **Note**
>
> Obwohl Sie Indizes jederzeit in Amazon DocumentDB erstellen können, ist es insgesamt schneller, Ihre Indizes zu erstellen, bevor Sie große Datensätze importieren. Als bewährte Methode empfehlen wir, dass Sie für jeden der folgenden Ansätze zuerst Ihre Indizes in Amazon DocumentDB erstellen, bevor Sie die Migration durchführen. Dazu können Sie das Amazon DocumentDB Index Tool verwenden.

Offline

Der Offline-Ansatz verwendet die `mongorestore` Tools `mongodump` und , um Ihre Daten von Ihrer MongoDB-Quellbereitstellung zu Ihrem Amazon DocumentDB-Cluster zu migrieren. Die Offline-Methode ist der einfachste Migrationsansatz, hat jedoch auch die längste Ausfallzeit für Ihren Cluster zur Folge.

Die grundlegende Vorgehensweise bei der Offline-Migration sieht wie folgt aus:

1. Stilllegen von Schreibvorgängen in Ihrer MongoDB-Quelle
2. Dump von Sammlungsdaten und Indizes aus der MongoDB-Quellenbereitstellung
3. Wenn Sie zu einem Elastic Cluster migrieren, erstellen Sie Ihre Sharded-Sammlungen mit dem `sh.shardCollection()` Befehl . Wenn Sie zu einem Instance-basierten Cluster migrieren, fahren Sie mit dem nächsten Schritt fort.
4. Stellen Sie Indizes im Amazon DocumentDB-Cluster wieder her.
5. Stellen Sie Sammlungsdaten im Amazon DocumentDB-Cluster wieder her.
6. Ändern Sie Ihren Anwendungsendpunkt, um in den Amazon DocumentDB-Cluster zu schreiben.



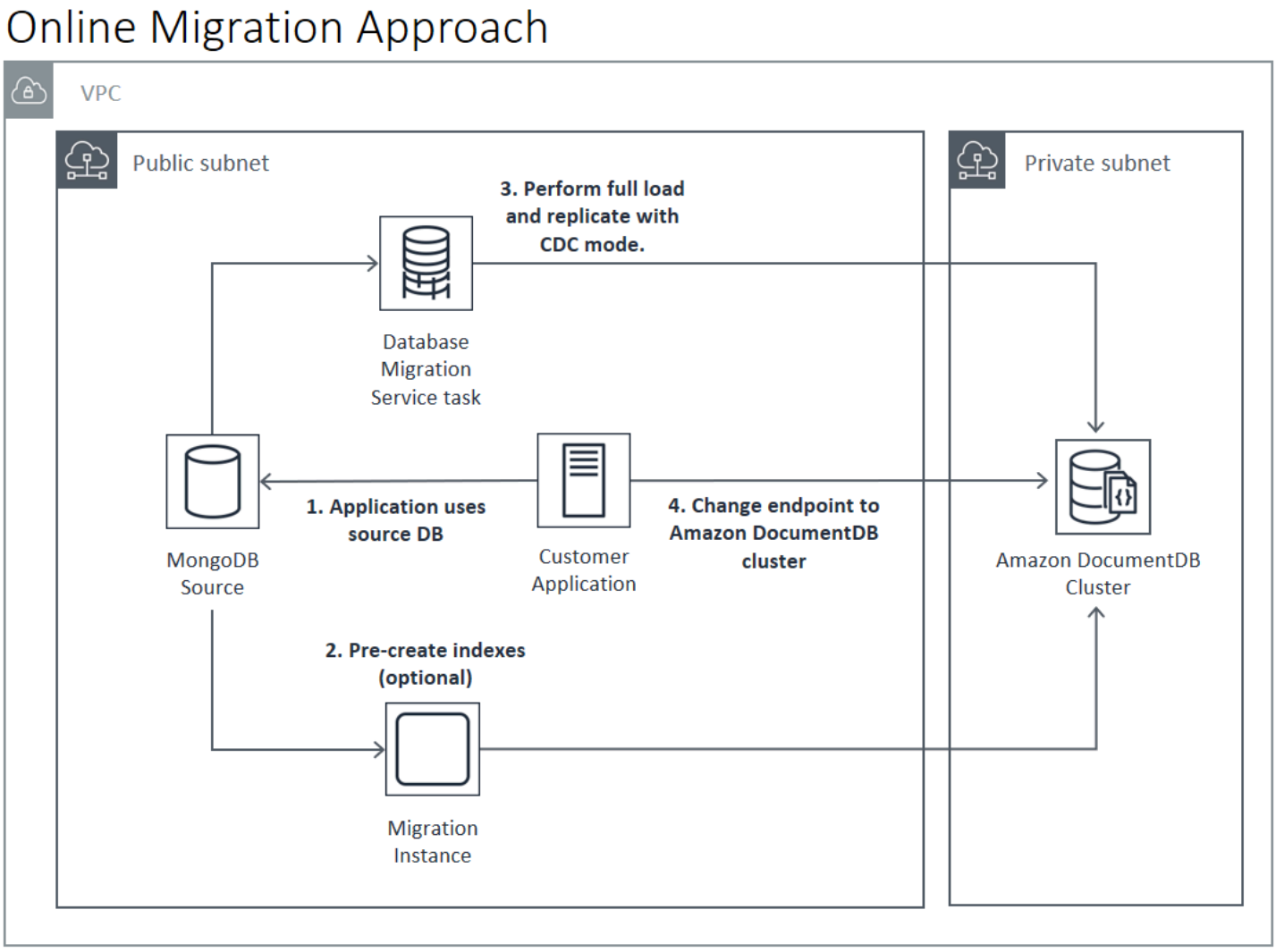
Status "Online"

Der Online- Ansatz verwendet AWS Database Migration Service (AWS DMS). Es führt ein vollständiges Laden von Daten aus Ihrer MongoDB-Quellbereitstellung in Ihren Amazon DocumentDB-Cluster durch. Danach findet ein Wechsel zum CDC-Modus (Change Data Capture) zum Replizieren von Änderungen statt. Der Onlineansatz minimiert die Ausfallzeiten für Ihren Cluster, ist aber die langsamste der drei Methoden.

Die grundlegende Vorgehensweise bei der Online-Migration sieht wie folgt aus:

1. Ihre Anwendung verwendet die Quell-DB normal.
2. Wenn Sie zu einem Elastic Cluster migrieren, erstellen Sie Ihre Sharded-Sammlungen mit dem `sh.shardCollection()` Befehl . Wenn Sie zu einem Instance-basierten Cluster migrieren, fahren Sie mit dem nächsten Schritt fort.

3. Erstellen Sie vorab Indizes im Amazon DocumentDB-Cluster.
4. Erstellen Sie eine -AWS DMSAufgabe, um einen vollständigen Ladevorgang durchzuführen, und aktivieren Sie dann CDC aus der MongoDB-Quellbereitstellung im Amazon DocumentDB-Cluster.
5. Nachdem die AWS DMS Aufgabe vollständig geladen wurde und Änderungen an Amazon DocumentDB repliziert hat, wechseln Sie den Endpunkt der Anwendung zum Amazon DocumentDB-Cluster.



Weitere Informationen zur Verwendung von AWS DMS für die Migration finden Sie unter Verwenden von Amazon DocumentDB als Ziel für AWS Database Migration Service und im zugehörigen Tutorial im AWS Database Migration Service -Benutzerhandbuch.

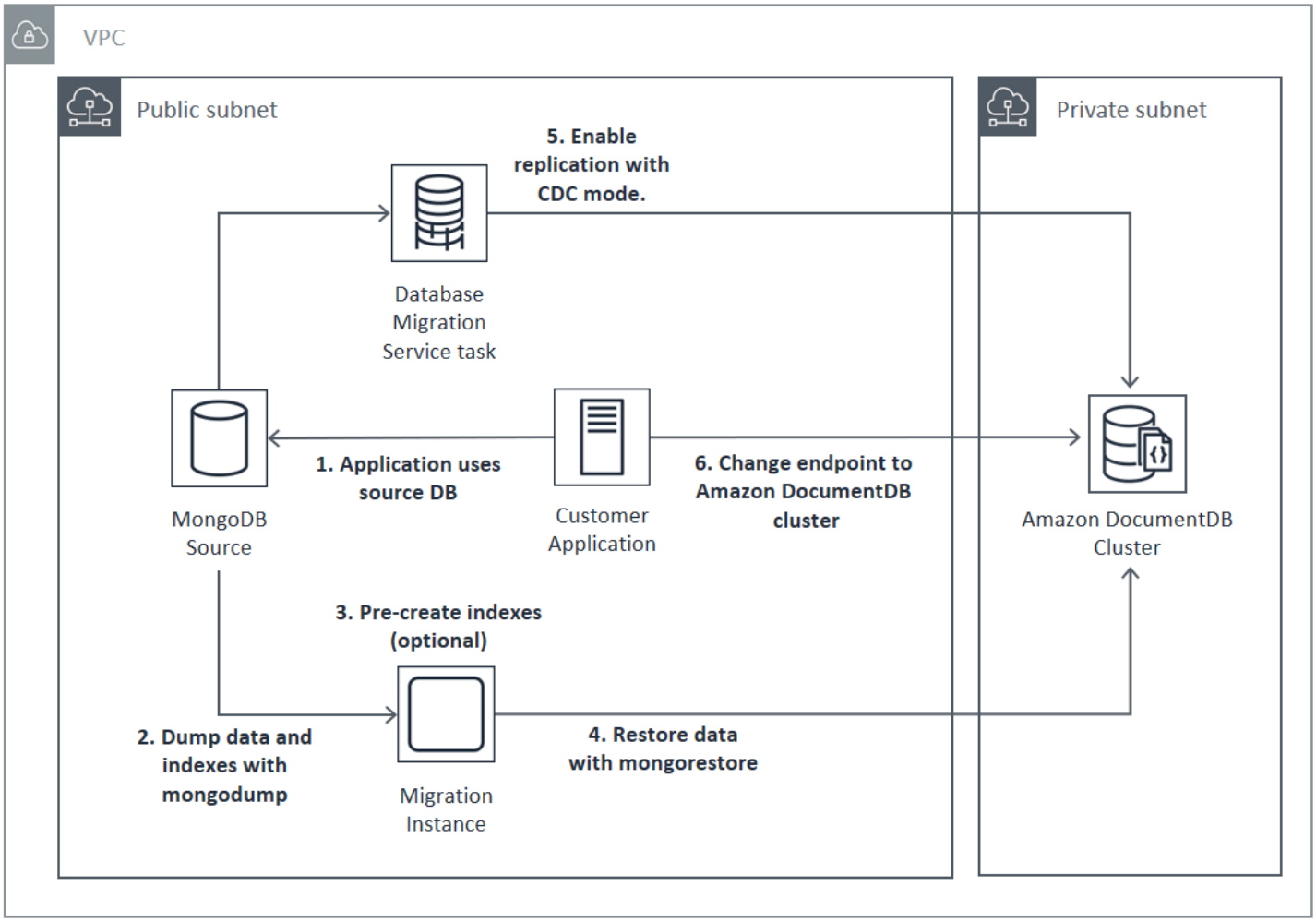
Hybrid

Der Hybrid-Ansatz verwendet die `mongorestore` Tools `mongodump` und , um Ihre Daten von Ihrer MongoDB-Quellbereitstellung zu Ihrem Amazon DocumentDB-Cluster zu migrieren. Im CDC-Modus wird dann AWS DMS zur Replikation von Änderungen verwendet. Der Hybrid-Ansatz ist ein Kompromiss zwischen Migrationsgeschwindigkeit und Ausfallzeit, ist jedoch auch der komplexeste der drei Ansätze.

Die grundlegende Vorgehensweise bei der Hybrid-Migration sieht wie folgt aus:

1. Ihre Anwendung verwendet die MongoDB-Quellenbereitstellung normal.
2. Dump von Sammlungsdaten und Indizes aus der MongoDB-Quellenbereitstellung
3. Stellen Sie Indizes im Amazon DocumentDB-Cluster wieder her.
4. Wenn Sie zu einem Elastic Cluster migrieren, erstellen Sie Ihre Sharded-Sammlungen mit dem `sh.shardCollection()` Befehl . Wenn Sie zu einem Instance-basierten Cluster migrieren, fahren Sie mit dem nächsten Schritt fort.
5. Stellen Sie Sammlungsdaten im Amazon DocumentDB-Cluster wieder her.
6. Erstellen Sie eine -AWS DMSAufgabe, um CDC aus der MongoDB-Quellbereitstellung im Amazon DocumentDB-Cluster zu aktivieren.
7. Wenn die AWS DMS Aufgabe Änderungen innerhalb eines akzeptablen Fensters repliziert, ändern Sie Ihren Anwendungsendpunkt so, dass er in den Amazon DocumentDB-Cluster schreibt.

Hybrid Migration Approach



> **Important**
>
> Eine AWS DMS-Aufgabe kann derzeit nur eine einzige Datenbank migrieren. Wenn Ihre MongoDB-Quelle über eine große Anzahl von Datenbanken verfügt, müssen Sie möglicherweise die Erstellung von Migrationsaufgaben automatisieren oder die Offlinemethode verwenden.

Unabhängig von dem von Ihnen gewählten Migrationsansatz ist es am effizientesten, Indizes in Ihrem Amazon DocumentDB-Cluster vorab zu erstellen, bevor Sie Ihre Daten migrieren. Dies liegt daran, dass Amazon DocumentDB-Indizes Daten parallel eingefügt werden, das Erstellen eines Index für vorhandene Daten jedoch eine Single-Thread-Operation ist.

Da AWS DMS keine Indizes migriert (nur die Daten), ist kein zusätzlicher Schritt erforderlich, um das erneute Anlegen von Indizes zu vermeiden.

Migrationsquellen

Wenn Ihre MongoDB-Quelle ein eigenständiger Mongo-Prozess ist und Sie die Online- oder Hybridmigration verwenden möchten, konvertieren Sie zunächst Ihren eigenständigen Mongo in einen Replikatsatz, sodass der Oplog für die Verwendung als CDC-Quelle erstellt wird.

Wenn Sie von einem MongoDB-Replikatsatz oder Sharded-Cluster migrieren, sollten Sie in Betracht ziehen, für jeden als Migrationsquelle verwendete Replikatsatz oder Shard einen verketteten oder versteckten Sekundärsatz zu erstellen. Die Durchführung von Daten-Dumps kann die Auslagerung von Daten aus dem Speicher bewirken und die Leistung von Produktions-Instances beeinträchtigen. Sie können dieses Risiko reduzieren, indem Sie von einem Knoten migrieren, der keine Produktionsdaten bereitstellt.

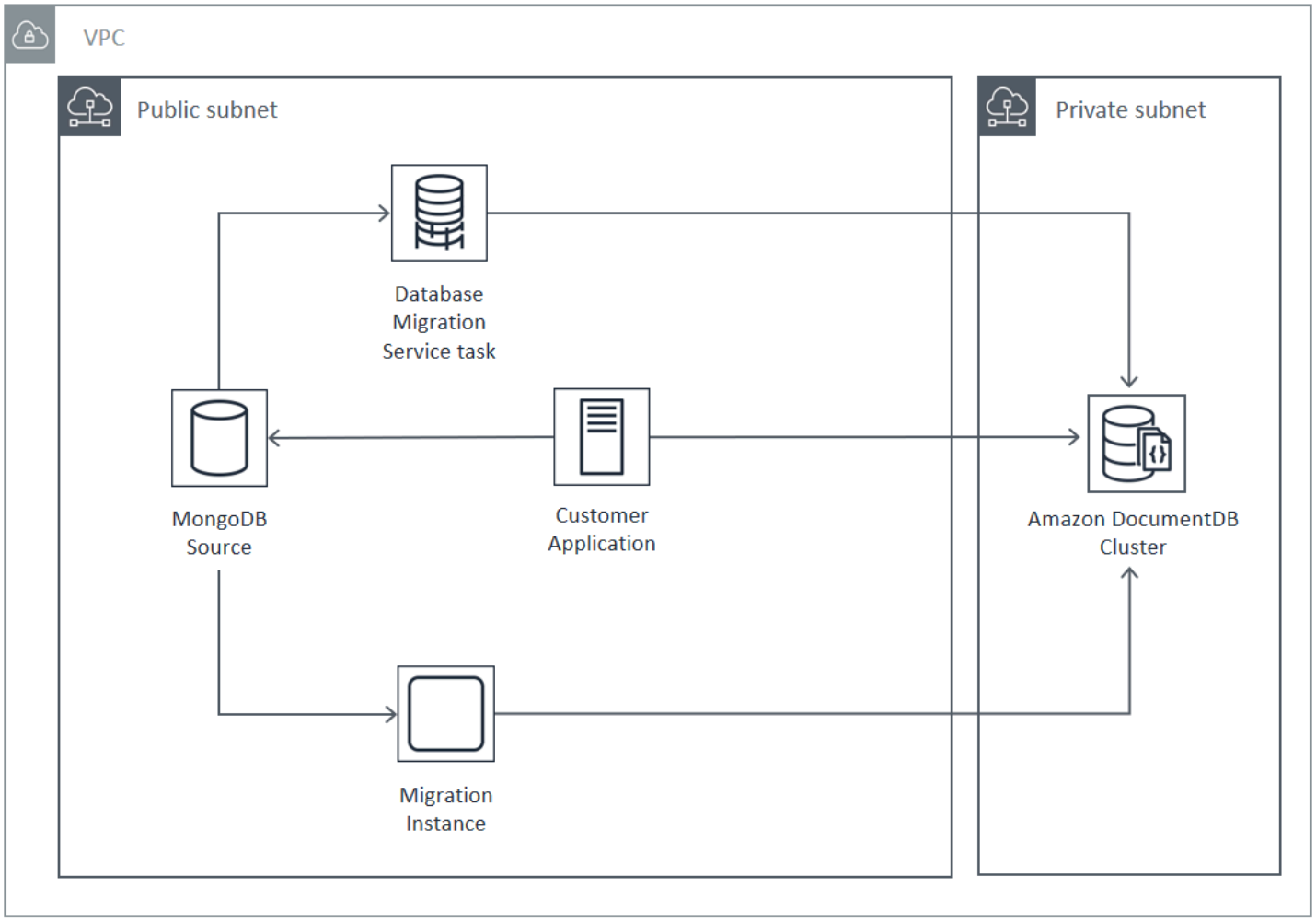
Version der Migrationsquelle

Wenn sich Ihre MongoDB-Quelldatenbankversion von der Kompatibilitätsversion Ihres Amazon DocumentDB-Ziel-Clusters unterscheidet, müssen Sie möglicherweise andere Vorbereitungsschritte unternehmen, um eine erfolgreiche Migration sicherzustellen. Die beiden häufigsten Anforderungen sind die Notwendigkeit, die MongoDB-Quellinstallation auf eine unterstützte Version für die Migration (MongoDB-Version 3.0 oder höher) zu aktualisieren und Ihre Anwendungstreiber zur Unterstützung der Amazon DocumentDB-Zielversion zu aktualisieren.

Wenn eine dieser Anforderungen vorliegt, sollten Sie diese Schritte zum Upgraden und Testen der Treiber in Ihren Migrationsplan aufnehmen.

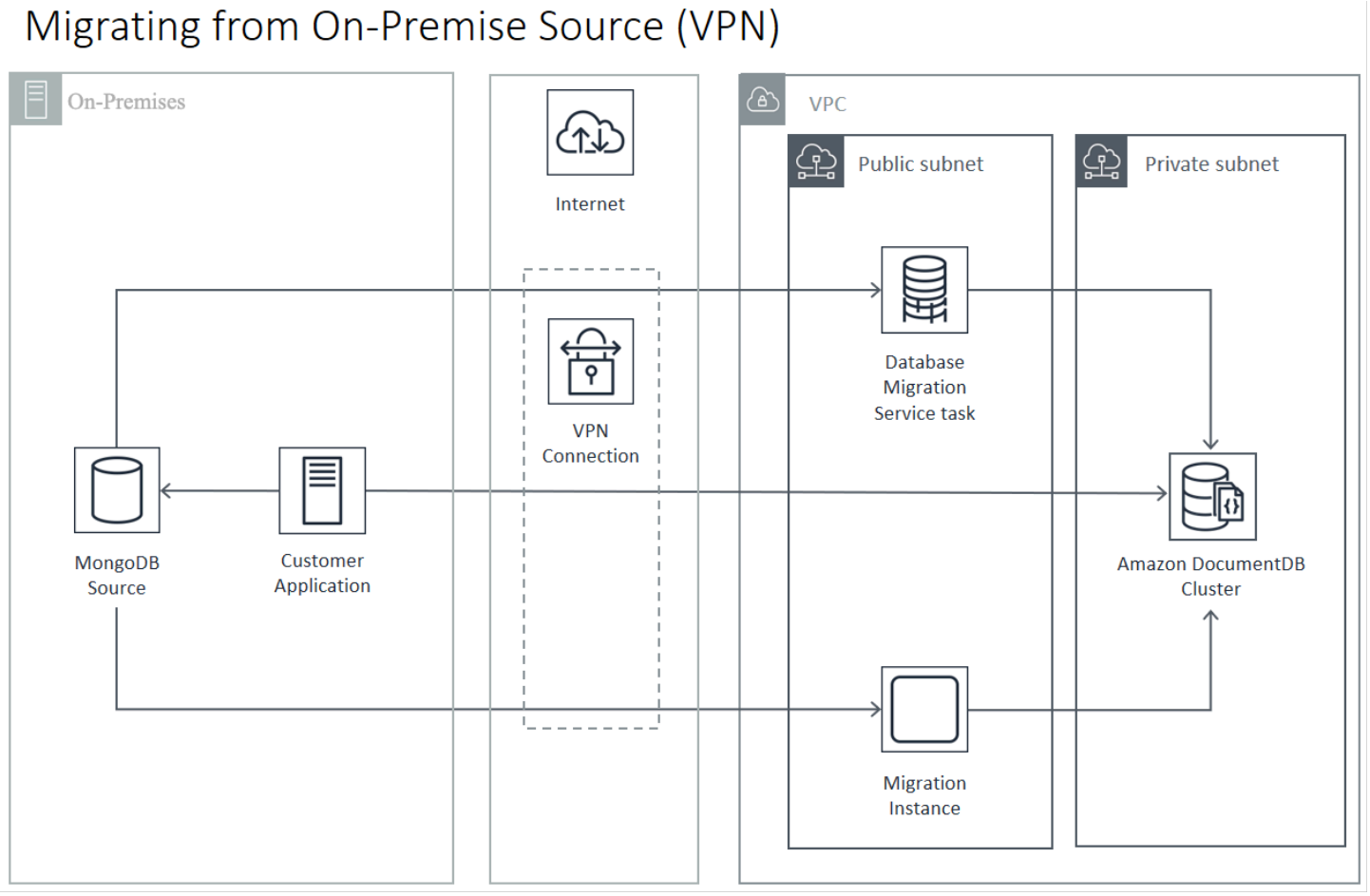
Konnektivität bei der Migration

Sie können von einer MongoDBAmazon DocumentDBQuellbereitstellung, die in Ihrem Rechenzentrum ausgeführt wird, oder von einer MongoDB-Bereitstellung, die auf einer Amazon EC2-Instance ausgeführt wird, zu Amazon DocumentDB migrieren. Die Migration von MongoDB zu EC2 ist einfach und erfordert nur, dass Sie Ihre Sicherheitsgruppen und Subnetze korrekt konfigurieren.

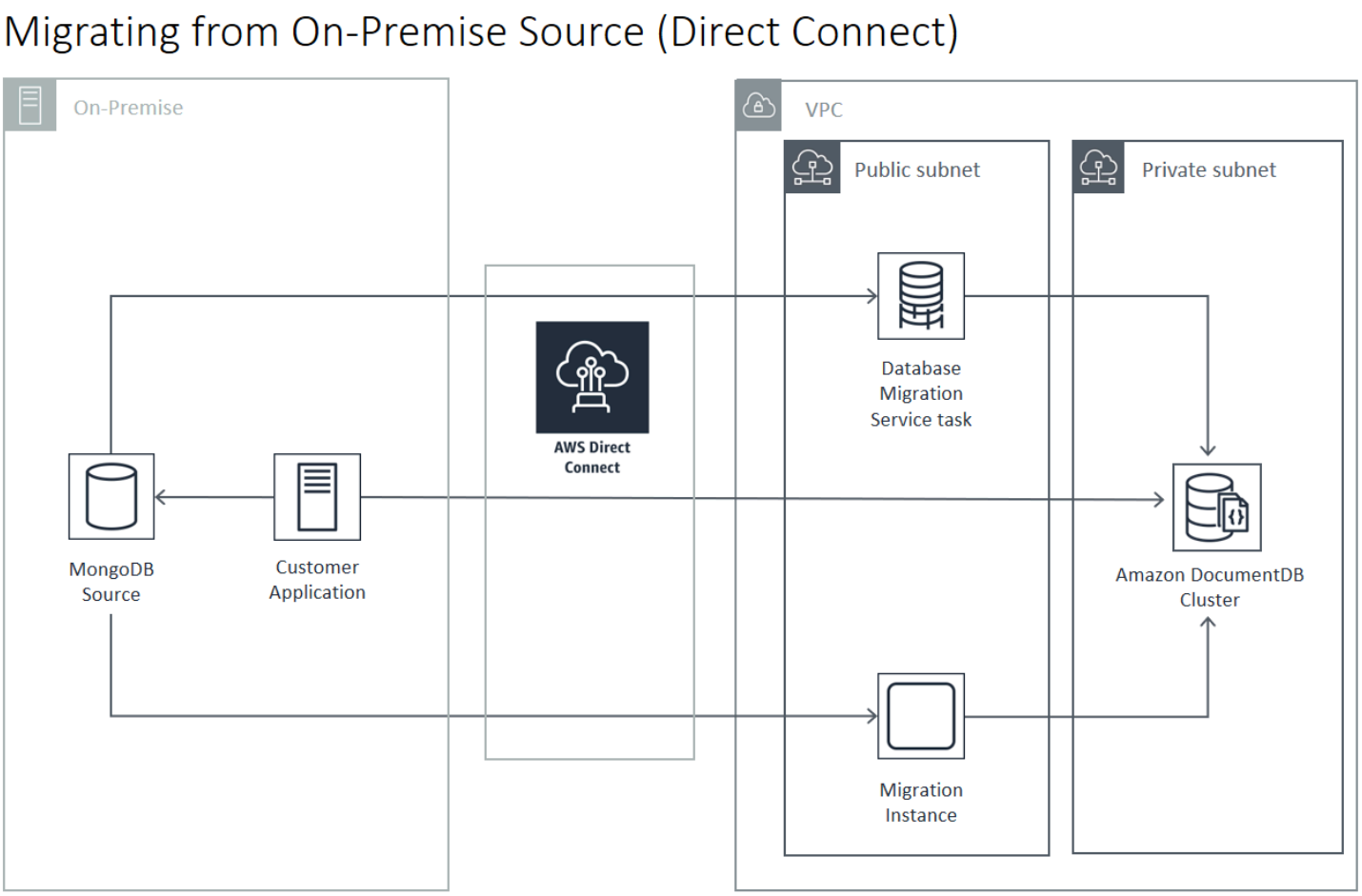


Die Migration aus einer Datenbank vor Ort erfordert eine Verbindung zwischen Ihrer MongoDB-Bereitstellung und Ihrer Virtual Private Cloud (VPC). Sie können dies über eine VPN-Verbindung (Virtual Private Network) oder über den AWS Direct Connect-Service erreichen. Obwohl Sie über das Internet zu Ihren VPC migrieren können, ist diese Verbindungsmethode aus Sicherheitsgründen am wenigsten wünschenswert.

Das folgende Diagramm veranschaulicht eine Migration zu Amazon DocumentDB von einer On-Premises-Quelle über eine VPN-Verbindung.



Im Folgenden wird eine Migration zu Amazon DocumentDB von einer On-Premises-Quelle mit dargestelltAWS Direct Connect.



Online- und Hybrid-Migrationsansätze erfordern die Verwendung einer InstanceAWS DMS, die auf Amazon EC2 in einer Amazon VPC ausgeführt werden muss. Alle Ansätze erfordern einen Migrationsserver für die Ausführung von `mongodump` und `mongorestore`. Es ist im Allgemeinen einfacher, den Migrationsserver auf einer Amazon EC2-Instance in der VPC auszuführen, in der Ihr Amazon DocumentDB-Cluster gestartet wird, da die Konnektivität zu Ihrem Amazon DocumentDB-Cluster erheblich vereinfacht wird.

Testen

Nachfolgend sind die Ziele des Pre-Migration-Tests aufgeführt:

- Überprüfen Sie, ob Ihr gewählter Ansatz das gewünschte Migrationsergebnis erzielt.
- Vergewissern Sie sich, dass der ausgewählte Instance-Typ und die Leseeinstellungsoptionen Ihren Anforderungen an die Anwendungsleistung entsprechen.
- Überprüfen Sie das Verhalten Ihrer Anwendung während des Failovers.

Migrationsplan – Überlegungen zum Test

Beachten Sie Folgendes, wenn Sie Ihren Amazon DocumentDB-Migrationsplan testen.

Themen

Wiederherstellen von Indizes

`mongorestore` erstellt standardmäßig Indizes für abgerufene Sammlungen. Dies geschieht jedoch erst nach der Wiederherstellung der Daten. Es ist insgesamt schneller, Indizes in Amazon DocumentDB zu erstellen, bevor Daten im Cluster wiederhergestellt werden. Der Grund dafür ist, dass die Indizierungsoperationen während des Ladens der Daten parallelisiert werden.

Wenn Sie sich für die Vorerrichtung Ihrer Indizes entscheiden, können Sie den Schritt der Indexerstellung bei der Wiederherstellung von Daten mit `mongorestore` überspringen, indem Sie die Option `--noIndexRestore` angeben.

Abrufen von Daten

Das `mongodump`-Tool ist die bevorzugte Methode, um Daten aus Ihrer MongoDB-Bereitstellung abzurufen. Abhängig von den Ressourcen, die für Ihre Migrations-Instance zur Verfügung stehen, können Sie `mongodump` beschleunigen, indem Sie mit der Option `--numParallelCollections` die Anzahl der parallelen Verbindungen (Standardwert: 4) erhöhen.

Wiederherstellen von Daten

Das `mongorestore` Tool ist die bevorzugte Methode zum Wiederherstellen von Dump-Daten auf Ihrer Amazon DocumentDB-Instance. Sie können die Wiederherstellungsleistung verbessern, indem Sie die Anzahl der Worker für jede Sammlung während der Wiederherstellung mit der Option `--numInsertionWorkersPerCollection` erhöhen. Ein Worker pro vCPU auf Ihrer primären Instance des Amazon DocumentDB-Clusters ist ein guter Ausgangspunkt.

Amazon DocumentDB unterstützt derzeit die `--oplogReplay` Option des `mongorestore` Tools nicht.

Standardmäßig überspringt `mongorestore` Einfügefehler und setzt den Wiederherstellungsprozess fort. Dies kann passieren, wenn Sie nicht unterstützte Daten auf Ihrer Amazon DocumentDB-Instance wiederherstellen. Es kann beispielsweise vorkommen, dass Sie ein Dokument haben, das Schlüssel oder Werte mit Null-Zeichenfolgen enthält. Wenn Sie es vorziehen, dass die Operation `mongorestore` vollständig fehlschlägt, wenn ein Wiederherstellungsfehler auftritt, verwenden Sie die Option `--stopOnError`.

Oplog-Dimensionierung

Das MongoDB-Operationsprotokoll (`oplog`) ist eine gedeckelte Sammlung, die alle Datenänderungen an Ihrer Datenbank enthält. Sie können die Größe des Oplogs und den darin enthaltenen Zeitraum anzeigen, indem Sie die Operation `db.printReplicationInfo()` für einen Replikatsatz oder ein Shard-Mitglied ausführen.

Wenn Sie die Online- oder Hybridansätze verwenden, stellen Sie sicher, dass das Oplog auf jedem Replikatsatz oder Shard groß genug ist, um alle während der gesamten Dauer des Datenmigrationsprozesses (egal, ob es sich um ein vollständiges Laden über `mongodump` oder eine AWS DMS-Aufgabe handelt) vorgenommenen Änderungen zu speichern. Es sollte außerdem einen angemessenen Puffer geben. Weitere Informationen finden Sie unter Überprüfen der Oplog-Größe in der MongoDB-Dokumentation. Bestimmen Sie die minimal erforderliche Oplog-Größe, indem Sie die verstrichene Zeit für den ersten Testlauf Ihres `mongodump`- oder `mongorestore`-Prozesses oder Ihrer AWS DMS-Aufgabe für das vollständige Laden erfassen.

AWS Database Migration Service-Konfiguration

Das AWS Database Migration Service -Benutzerhandbuch behandelt die Komponenten und Schritte, die für die Migration Ihrer MongoDB-Quelldaten zu Ihrem Amazon DocumentDB-Cluster erforderlich sind. Im Folgenden wird der grundlegende Prozess für die Verwendung von AWS DMS zur Durchführung einer Online- oder Hybridmigration beschrieben:

So führen Sie eine Migration mit AWS DMS durch:

1. Erstellen Sie einen MongoDB-Quellendpunkt. Weitere Informationen finden Sie unter Verwendung von MongoDB als Quelle für AWS DMS.
2. Erstellen Sie einen Amazon DocumentDB-Zielendpunkt. Weitere Informationen finden Sie unter Arbeiten mit AWS DMS-Endpunkten.

Wenn Sie Ihren Zielendpunkt als elastischen Cluster konfigurieren, beachten Sie, dass Ihr vorhandenes Amazon DocumentDB-SSL-Zertifikat nicht mit elastischen Clustern funktioniert und Sie Ihrem Endpunkt mithilfe der folgenden Schritte ein neues SSL-Zertifikat anfügen müssen:

a. Besuchen Sie https://www.amazontrust.com/repository/SFSRootCAG2.pem und speichern Sie den Inhalt als „SFSRootCAG2.pem”Datei. Dies ist die Zertifikatdatei, die Sie in nachfolgenden Schritten importieren müssen.

b. Wählen Sie beim Erstellen des Elastic-Cluster-Endpunkts unter Endpunktkonfiguration die Option Neues CA-Zertifikat hinzufügen aus.

- Geben Sie für Zertifikat-ID `SFSRootCAG2.pem` ein.
- Wählen Sie unter Import certificate file (Zertifikatsdatei importieren) die Option Choose file (Datei auswählen) und navigieren Sie zu der `SFSRootCAG2.pem`-Datei, die Sie zuvor heruntergeladen haben. Wählen Sie die Datei aus und öffnen Sie sie. Wählen Sie Zertifikat importieren und dann `SFSRootCAG2.pem` aus der Dropdownliste Zertifikat auswählen aus.

3. Erstellen Sie mindestens eine AWS DMS-Replikations-Instance. Weitere Informationen finden Sie unter Arbeiten mit einer AWS DMS-Replikations-Instance.
4. Erstellen Sie mindestens eine AWS DMS-Replikationsaufgabe. Weitere Informationen finden Sie unter Arbeiten mit AWS DMS-Aufgaben.

   Für eine Onlinemigration verwendet Ihre Migrationsaufgabe den Migrationstyp Migrate existing data and replicate ongoing changes (Migrieren von bestehenden Daten und Replizieren nachfolgender Änderungen).

   Für eine Hybridmigration verwendet Ihre Migrationsaufgabe den Migrationstyp Replicate data changes only (Nur Datenänderungen replizieren). Sie können die CDC-Startzeit so wählen, dass sie mit dem Dump-Zeitpunkt aus Ihrer `mongodump`-Operation übereinstimmt. Das MongoDB-Oplog ist idempotent. Um fehlende Änderungen zu vermeiden, ist es ratsam, einige Minuten Überlappung zwischen Ihrer `mongodump`-Endzeit und Ihrer CDC-Startzeit zu lassen.

Migration von einem Sharded-Cluster aus

Der Prozess für die Migration von Daten von einem MongoDB-Sharded-Cluster zu Ihrer Amazon DocumentDB-Instance ist im Wesentlichen der mehrerer parallel ausgeführter Replikatsatzmigrationen. Ein wichtiger Aspekt beim Testen einer Migration von Sharded-Clustern ist, dass einige Shards möglicherweise stärker genutzt werden als andere. Diese Situation führt

zu unterschiedlichen Laufzeiten der Datenmigration. Stellen Sie sicher, dass Sie die `oplog` Anforderungen jedes Shards bei der Planung und dem Testen bewerten.

Im Folgenden sind einige Konfigurationsprobleme aufgeführt, die bei der Migration eines Sharded-Clusters zu berücksichtigen sind:

- Bevor Sie `mongodump` ausführen oder eine AWS DMS-Migrationsaufgabe starten, müssen Sie den Sharded-Cluster-Balancer deaktivieren und warten, bis alle laufenden Migrationen abgeschlossen sind. Weitere Informationen finden Sie unter Balancer deaktivieren in der MongoDB-Dokumentation.
- Wenn Sie AWS DMS zur Datenreplikation verwenden, führen Sie den Befehl `cleanupOrphaned` auf jedem Shard aus, bevor Sie die Migrationsaufgaben ausführen. Wenn Sie diesen Befehl nicht ausführen, können die Aufgaben aufgrund doppelter Dokument-IDs fehlschlagen. Beachten Sie, dass dieser Befehl die Leistung beeinträchtigen kann. Weitere Informationen finden Sie unter cleanupOrphaned in der MongoDB-Dokumentation.
- Wenn Sie das `mongodump`-Tool zum Abrufen von Daten verwenden, sollten Sie einen `mongodump`-Prozess pro Shard ausführen. Der zeiteffizienteste Ansatz erfordert möglicherweise mehrere Migrationsserver, um die Leistung Ihres Dumps zu maximieren.
- Wenn Sie Daten mit AWS Database Migration Service replizieren, müssen Sie für jeden Shard einen Quellendpunkt anlegen. Führen Sie außerdem mindestens eine Migrationsaufgabe für jeden Shard aus, den Sie migrieren. Der zeiteffizienteste Ansatz erfordert möglicherweise mehrere Replikations-Instances, um die Migrationsleistung zu maximieren.

Leistungstests

Nachdem Sie Ihre Daten erfolgreich zu Ihrem Amazon DocumentDB-Testcluster migriert haben, führen Sie Ihren Test-Workload für den Cluster aus. Stellen Sie mithilfe von Amazon- CloudWatch Metriken sicher, dass Ihre Leistung den aktuellen Durchsatz Ihrer MongoDB-Quellbereitstellung erfüllt oder überschreitet.

Überprüfen Sie die folgenden wichtigen Amazon DocumentDB-Metriken:

- Netzwerkdurchsatz
- Schreibdurchsatz
- Lesedurchsatz
- Replikatverzögerung

Weitere Informationen finden Sie unter Amazon-Dokumententententent.

Failover-Tests

Stellen Sie sicher, dass das Verhalten Ihrer Anwendung während eines Failover-Ereignisses von Amazon DocumentDB Ihre Verfügbarkeitsanforderungen erfüllt. Um ein manuelles Failover eines Amazon DocumentDB-Clusters auf der Konsole zu initiieren, wählen Sie auf der Seite Cluster die Failover-Aktion im Menü Aktionen aus.

Sie können einen Failover auch einleiten, indem Sie die Operation `failover-db-cluster` von der AWS CLI aus ausführen. Weitere Informationen finden Sie unter `failover-db-cluster` im Abschnitt Amazon DocumentDB der -AWS CLIReferenz.

Weitere Ressourcen

Weitere Informationen finden Sie in den folgenden Themen im AWS Database Migration Service-Benutzerhandbuch:

- Verwenden von Amazon DocumentDB als Ziel für AWS Database Migration Service
- Anleitung: Migration von MongoDB nach Amazon DocumentDB

Migrations-Playbook: MongoDB zu Amazon DocumentDB

Dieses Migrations-Playbook bietet Ihnen Ressourcen und Schritte, die Sie bei der Migration von einer MongoDB-Datenbank zu Amazon DocumentDB unterstützen.

Migrationsprozess

Im Folgenden sind die allgemeinen Schritte aufgeführt, die in der Regel bei der Migration Ihrer Daten von einer MongoDB-Datenbank zu Amazon DocumentDB erforderlich sind.

Themen

- Schritt 1: Kompatibilität und funktionale Unterschiede
- Schritt 2: Machbarkeitsnachweis
- Schritt 3: Migrieren der Daten
- Schritt 4: Datenvalidierung
- Schritt 5: Anwendungs-Cutover

Schritt 1: Kompatibilität und funktionale Unterschiede

Amazon DocumentDB interagiert mit den Apache-2.0-Open-Source-APIs MongoDB 3.6, 4.0 und 5.0 APIs. Daher können Sie dieselben MongoDB-Treiber, -Anwendungen und -Tools mit Amazon DocumentDB ohne oder ohne Änderungen verwenden.

Der erste Schritt besteht darin, die Kompatibilität zwischen den Operatoren und Indizes zu überprüfen, die Ihre Anwendung in Ihrer MongoDB-Datenbank verwendet, und deren Verfügbarkeit in Amazon DocumentDB zu verstehen und die funktionalen Unterschiede zwischen ihnen zu verstehen.

Kompatibilität von Operatoren

Verwenden Sie das Kompatibilitätstool von Amazon DocumentDB *, um leicht zu erkennen, ob Ihre Anwendung in ihren Abfragen nicht unterstützte Operatoren verwendet. Dieses Tool kann Ihre MongoDB-Datenbankserver-Protokolldateien oder Ihren Anwendungsquellcode scannen, um einen Bericht über nicht unterstützte Operatoren bereitzustellen. Wenn Sie feststellen, dass nicht unterstützte Operatoren verwendet werden, müssen Sie Ihre Anwendung so ändern, dass sie nicht unterstützte Operatoren umgeht.

Führen Sie die folgenden Schritte aus, um die Kompatibilität zwischen den in Ihrem Setup verwendeten MongoDB-Operatoren und den unterstützten Amazon DocumentDB-Operatoren zu überprüfen:

```
git clone https://github.com/awslabs/amazon-documentdb-tools.git
cd amazon-documentdb-tools/compat-tool/
python3 compat.py --version <Amazon DocumentDB version> --directory <mongodb logfiles/
source code>
```

Weitere Informationen finden Sie unter Unterstützte MongoDB-APIs, -Operationen und -Datentypen.

* Wird nicht offiziell von unterstütztAWS.

Kompatibilität von Indizes

Sie können das Amazon DocumentDB-Indextool * verwenden, um herauszufinden, ob Sie Indextypen verwenden, die in Amazon DocumentDB nicht unterstützt werden. Dieses Tool benötigt eine Verbindung zu Ihrer Quelldatenbank, um Indexdefinitionen zu lesen.

Dazu müssen Sie zunächst Indexdefinitionen mit der `--dump-indexes` Option in ein Verzeichnis ablegen. Führen Sie dann das Tool mit der `--show-issues` Option aus und stellen Sie das Verzeichnis bereit, in dem Sie inkompatible Indizes finden können.

Exportieren von Indizes:

```
git clone https://github.com/awslabs/amazon-documentdb-tools.git
sudo pip install -r amazon-documentdb-tools/index-tool/requirements.txt
mkdir <directory to dump index definitions>
python3 migrationtools/documentdb_index_tool.py --dump-indexes --dir <directory> --uri
 <source-mongodb-uri>
```

Suchen Sie nach inkompatiblen Indizes:

```
python3 migrationtools/documentdb_index_tool.py --show-issues --dir <dumped-index-
definitions-directory>
```

Wenn Sie feststellen, dass nicht unterstützte Indextypen verwendet werden, müssen Sie Ihre Anwendung oder Ihr Datenmodell ändern, um ohne die inkompatiblen Indizes zu umgehen oder fortzufahren.

Weitere Informationen zu unterstützten Indextypen und Eigenschaften in Amazon DocumentDB finden Sie unter Indizes und Indexeigenschaften und So indizieren Sie in Amazon DocumentDB.

* Wird nicht offiziell von unterstütztAWS.

Funktionsunterschiede

Lesen Sie Funktionale Unterschiede zu MongoDB, um sich mit den Unterschieden vertraut zu machen.

Schritt 2: Machbarkeitsnachweis

Führen Sie einen Machbarkeitsnachweis durch, indem Sie Ihre Anwendung oder Ihre reguläre Testsuite auf Amazon DocumentDB ausführen, um die Funktionalität und Leistung zu testen. Möglicherweise müssen Sie Ihren Amazon DocumentDB-Cluster mit Daten füllen, um die Tests durchführen zu können. Sie können beispielsweise die `mongorestore` Tools `mongodump` und verwenden, um Daten aus Ihrer MongoDB-Quell- zu kopieren.

Funktionstests

Erstellen Sie einen Amazon DocumentDB-Cluster (siehe Erstellen eines Amazon DocumentDB-Clusters) und führen Sie Ihre Anwendung oder Ihre funktionale Testsuite aus, um zu überprüfen, ob alle Anwendungsworkflows weiterhin reibungslos auf Amazon DocumentDB funktionieren.

Leistungstests

Führen Sie Leistungstests für Ihre Anwendung oder Ihre Leistungstestsuite durch, die auf Amazon DocumentDB ausgeführt wird, mit einem Workload, der Ihrem Produktions-Workload ähnelt, um festzustellen, ob die Einrichtung Ihren Latenzanforderungen entspricht. Optimieren Sie Ihren Workload auf Leistung oder skalieren Sie Ihren Amazon DocumentDB-Cluster nach Bedarf. Weitere Informationen finden Sie unter Leistung und Ressourcenauslastung und Skalieren von Amazon DocumentDB-Clustern.

Es ist wichtig, die Größe Ihres Amazon DocumentDB-Clusters mit den richtigen Instance-Typen für eine optimale Leistung zu erhöhen. Weitere Informationen finden Sie unter Bewährte Methoden für Instance-Größenbestimmung.

Sie können den Größenrechner * von Amazon DocumentDB verwenden, um Ihnen bei der Schätzung der Größe Ihres Amazon DocumentDB-Clusters zu helfen.

* Wird von nicht offiziell unterstütztAWS.

Failover-Tests

Möglicherweise möchten Sie beobachten, wie Ihre Anwendung auf einen Neustart des Amazon DocumentDB-Primärknotens, ein Failover des Primärknotens oder eine Löschung des Primärknotens in einem Cluster mit mehreren Knoten reagiert, sowie wenn Replikatknoten neu gestartet oder entfernt werden. Auf diese Weise können Sie überprüfen, ob Ihre Anwendung gegenüber diesen Ereignissen ausfallsicher ist. Weitere Informationen finden Sie unter Testen eines Failovers.

Informationen zu den Ausnahmen, die eine Anwendung tolerieren sollte, und zu deren effizienter Handhabung finden Sie unter Erstellen ausfallsicherer Anwendungen mit Amazon DocumentDB.

> **Note**
>
> Es gibt keinen Ersatz für das Testen Ihrer Workload auf Amazon DocumentDB

Schritt 3: Migrieren der Daten

Migrieren Sie nach einem erfolgreichen Machbarkeitsnachweis Ihre Daten zu Amazon DocumentDB . Die meisten unserer Kunden verwenden Online- oder Offline-Migrationsansätze, um ihre Daten zu migrieren.

Online-Migration

Mit der Online-Migrationsmethode können Sie Daten von Ihrer Quelldatenbank, die von einigen Gigabyte bis hin zu mehreren Terabyte reichen, mit nahezu null Ausfallzeiten zu Amazon DocumentDB migrieren. Weitere Informationen finden Sie unter AWS Database Migration Service (AWS DMS).

Wenn Sie von einer MongoDB-Datenbank migrieren, können Sie verwendenAWS DMS, um einen vollständigen Ladevorgang durchzuführen und laufende Änderungen zu replizieren.

Einen step-by-step Prozess finden Sie unter Migrieren zu Amazon DocumentDB mit der Online-Methode .

Weitere Informationen finden Sie im Abschnitt Verwenden von Amazon DocumentDB als Ziel für AWS Database Migration Service im AWS Database Migration Service -Benutzerhandbuch.

Punkte, die Sie mit beachten solltenAWS DMS:

- Segmentierung: Bei der Migration von Multi-Terabyte-AWS DMSDatenbanken mit kann es mit den Standardeinstellungen langsam sein, da die Volllast von DMS standardmäßig Single-Thread pro Sammlung ist, was zu längeren Migrationszeiten führt. Um die Volllast bei großen Datenbankmigrationen zu beschleunigen, können Sie die Segmentierungsfunktion in verwendenAWS DMS.

  Weitere Informationen zur Verwendung der Segmentierung mit AWS DMSfinden Sie unter Verwenden der automatischen Segmentierung mit AWS DMS.
- DMS-Instance-Typ: Um die Datenmigration zu beschleunigen, müssen Sie die richtige DMS-Instance auswählen.

Offline-Migration

Die Offline-Migration ist der einfachste Ansatz, um Datenbanken zu Amazon DocumentDB zu verschieben. Dieser Ansatz wird hauptsächlich für POCs und für Workloads verwendet, die während der Migration Schreibausfallzeiten verursachen können.

Einen step-by-step Prozess finden Sie unter Migrieren von MongoDB zu Amazon DocumentDB mit der Offline-Methode .

Schritt 4: Datenvalidierung

Sobald die Daten erfolgreich migriert wurden, überprüfen Sie die Daten auf ihre Richtigkeit, um Vertrauen zu gewinnen. In der Konsole der AWS DMS Migrationsaufgabe finden Sie migrierte Datenmetriken. Weitere Informationen finden Sie unter Überprüfen migrierter Daten.

Sie können auch das Amazon DocumentDB DataDiffer -Tool * verwenden, um die Datenkonsistenz zwischen den Quell- und Zielsammlungen zu überprüfen.

* Offiziell nicht von unterstütztAWS.

Schritt 5: Anwendungs-Cutover

Dazu müssen Sie die Datenbankverbindungszeichenfolge Ihrer Anwendung ändern, um Ihren Amazon DocumentDB-Cluster zu verwenden.

Weitere Informationen zum Herstellen einer Verbindung mit Amazon DocumentDB finden Sie unter Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz.

Online-Migration

Nachdem der vollständige Datenladevorgang abgeschlossen ist, repliziert AWS DMS weiterhin laufende Änderungen von Ihrer Quelle auf Amazon DocumentDB . Nachdem die Änderungen abgeschlossen sind und Ihre Datenvalidierungsprüfungen abgeschlossen sind, können Sie einen Cutover auf Amazon DocumentDB durchführen.

Offline-Migration

Nach Abschluss der vollständigen Datenlade- und Datenvalidierungsprüfungen können Sie den Cutover auf Amazon DocumentDB durchführen.

Weitere Ressourcen

Hier sind einige zusätzliche Ressourcen, die bei Ihrer Migration helfen könnten:

- Video: Bewährte Methoden für die Migration zu Amazon DocumentDB
- Video: Erste Schritte mit Amazon DocumentDB Observability and Monitoring
- Zusätzliche Dienstprogramme: Amazon DocumentDB-Tools*
- Migrationsentwicklerhandbuch: Migration zu Amazon DocumentDB

* Offiziell nicht von unterstütztAWS.

Direktes Upgrade der Hauptversion von Amazon DocumentDB

Amazon DocumentDB stellt neue Versionen von Datenbank-Engines im Allgemeinen erst nach umfangreichen Tests zur Verfügung. Sie können wählen, wie und wann Sie Ihre Amazon DocumentDB-Cluster auf die neue Version aktualisieren möchten.

Derzeit unterstützt Amazon DocumentDB drei Hauptversionen: Amazon DocumentDB 3.6, 4.0 und 5.0. Sie können ein direktes Hauptversions-Upgrade (MVU) Ihrer Datenbank durchführen und gleichzeitig dieselben Endpunkte, Speicher und Tags der Cluster beibehalten und Ihre Anwendungen ohne Änderungen weiter verwenden. Diese Funktion ist in allen Regionen, in denen Amazon DocumentDB 5.0 verfügbar ist, kostenlos verfügbar.

> **Important**
>
> Ihre Amazon DocumentDB-Cluster sind während des direkten Hauptversions-Upgrades nicht verfügbar und bei Ihren Clustern treten mehrere Neustarts auf. Die Upgrade-Ausfallzeit kann je nach Anzahl der Sammlungen, Indizes, Datenbanken und Instances von Cluster zu Cluster variieren. Wir empfehlen, das Upgrade während Ihres Wartungsfensters oder während Zeiten geringer Auslastung durchzuführen. Sobald Ihr Cluster aktualisiert wurde, können Sie den Cluster nicht auf eine frühere Version herunterstufen, aber Sie können Ihren Snapshot vor dem Upgrade auf einem neuen Cluster wiederherstellen.

Themen

Voraussetzungen und Einschränkungen

Im Folgenden sind die Voraussetzungen und Einschränkungen für das direkte Upgrade von Hauptversionen aufgeführt, die Sie möglicherweise verstehen und entsprechend handeln müssen, bevor Sie das Upgrade durchführen:

- Instance-Typ – Amazon DocumentDB 4.0/5.0 unterstützt keine r4.*-Instances. Um mit einem direkten Upgrade der Hauptversion fortzufahren, ändern Sie r4.*-Instances in r5.*-Instances. Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB-Instance. Informationen zu unterstützten Unterstützte Instance-Klassen nach Region Instances, die auf der Amazon DocumentDB-Engine-Version basieren, finden Sie unter .
- Instance-Betriebssystem-Patches – Ein direktes Upgrade der Hauptversion benötigt den neuesten Betriebssystem-Patch (OS), um fortzufahren. Bitte wenden Sie alle ausstehenden Betriebssystemwartungsaktionen auf die Instances an, bevor Sie mit dem direkten Upgrade fortfahren. Weitere Informationen finden Sie unter Arbeiten mit Betriebssystem-Updates.

  > **Note**
  >
  > In einigen Situationen sind Instance-Betriebssystem-Patches nicht sichtbar, wenn Sie ausstehende Engine-Patches auf Clusterebene haben. Möglicherweise müssen Sie Engine-Patches auf Clusterebene anwenden, bevor Sie mit der Anwendung von Instance-Betriebssystem-Patches und anschließend dem direkten Upgrade der Hauptversion fortfahren. Siehe Durchführen eines Patch-Updates für die Engine-Version eines Clusters.

- Das direkte Upgrade der Hauptversion ist in allen Regionen verfügbar, in denen Amazon DocumentDB 5.0 verfügbar ist.
- Das direkte Upgrade der Hauptversion wird mit Amazon DocumentDB 4.0 als Zielversion nicht unterstützt.
- Das direkte Upgrade der Hauptversion wird derzeit auf globalen Amazon DocumentDB-Clustern und elastischen Clustern nicht unterstützt.

  > **Note**
  >
  > Um Ihre globalen Cluster zu aktualisieren, löschen Sie Ihre sekundären Cluster aus dem globalen Cluster, konvertieren Sie den primären Cluster in einen regionalen Cluster, führen Sie ein direktes Hauptversions-Upgrade auf dem regionalen (primären) Cluster durch und erstellen Sie dann den globalen Cluster neu, indem Sie sekundäre Cluster mit demselben

Namen hinzufügen, um dieselben Endpunkte wie zuvor beizubehalten. Beachten Sie, dass E/A-Gebühren anfallen, während Ihr aktualisierter primärer Cluster Daten auf Ihre neu hinzugefügten sekundären Cluster repliziert. Ausführliche Schritte zum Entfernen sekundärer Cluster aus dem globalen Cluster vor dem Löschen finden Sie unter Entfernen eines Clusters aus einem globalen Amazon DocumentDB-Cluster.

- Wenn Sie eine große Anzahl von Indizes (>10.000) haben und auf einer kleineren Instance (z. B. t3.medium) arbeiten, müssen Sie Ihre primäre Instance auf eine größere Instance (z. B. mindestens r5.xlarge) hochskalieren, um genügend Speicher in der Instance zu reservieren, um das direkte Upgrade der Hauptversion durchzuführen. Sie können wählen, ob Sie die Instance-Größe herunterskalieren möchten, sobald Ihr direktes Upgrade der Hauptversion abgeschlossen ist. In den folgenden Tabellen finden Sie die maximale Anzahl von Indizes, die auf jedem Instance-Typ für ein direktes Upgrade der Hauptversion unterstützt werden:

  Für speicheroptimierte Instances (db.r5.*):

Instance	Maximale unterstützte Indizes für In-Place-MVU
db.r5.large	100K
db.r5.xlarge	200K
db.r5.2xlarge	300K
db.r5.4xlarge	400K
db.r5.8xlarge	500K
db.r5.12xlarge	700K
db.r5.16xlarge	800K
db.r5.24xlarge	1M

  Für Instances mit Spitzenlastleistung (db.t3, db.t4g)

Instance	Maximale unterstützte Indizes für In-Place-MVU
db.t4g.medium	3K
db.t3.medium	10 K

Für arbeitsspeicheroptimierte ton-Instances (db.r6g.*):

Instance	Maximale unterstützte Indizes für In-Place-MVU
db.r6g.large	100K
db.r6g.xlarge	200K
db.r6g.2xlarge	300K
db.r6g.4xlarge	400K
db.r6g.8xlarge	500K
db.r6g.12xlarge	700K
db.r6g.16xlarge	800K

> **Note**
>
> Wenn Sie mehr als 1M-Indizes haben, wenden Sie sich bitte an den AWS Support und fahren Sie nicht mit einem direkten Hauptversions-Upgrade fort.

Bewährte Methoden für direkte Hauptversions-Upgrades

Testen Sie direkte Hauptversions-Upgrades mit geklonten Clustern

1. Um direkte Hauptversions-Upgrades zu testen, empfehlen wir, die Funktion zum schnellen Klonen zu verwenden, um einen Klon Ihres Ziel-Clusters zu erstellen. Es fallen keine Speicherkosten für das Testen des direkten Hauptversions-Upgrades auf einem geklonten Volume an, es sei denn, Sie ändern Daten auf dem Cluster. Weitere Informationen zum Volume-Klon finden Sie unter Klonen eines Volumes für einen Amazon DocumentDB-Cluster.
2. Um eine realistischere Schätzung der Zeit zu erhalten, die für den Abschluss des direkten Hauptversions-Upgrades benötigt wird, passen Sie die Instance-Anzahl des geklonten Clusters dem Ziel-Cluster an.
3. Wir empfehlen, den neu aktualisierten Amazon DocumentDB-5.0-Cluster vollständig auf funktionale Unterschiede zu testen, um sicherzustellen, dass alles wie erwartet funktioniert.

Vor einem direkten Upgrade der Hauptversion

1. Halten Sie eine versionskompatible Cluster-Parametergruppe bereit.

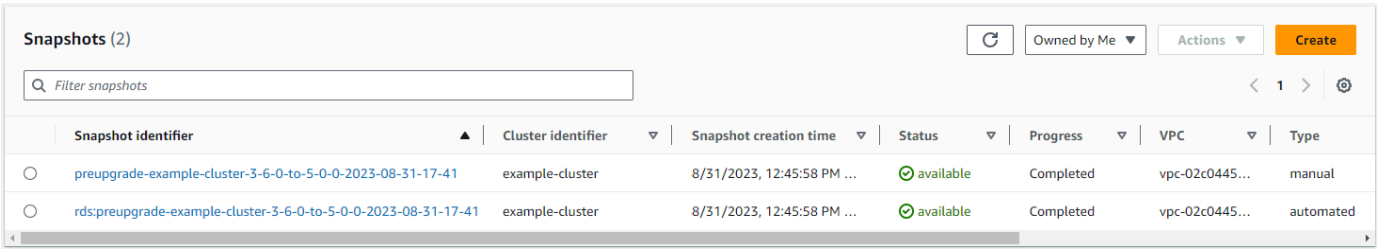
   Verwenden Sie die Amazon DocumentDB-Standard-Cluster-Parametergruppe für die neue Engine-Version oder erstellen Sie Ihre eigene benutzerdefinierte Cluster-Parametergruppe für die neue Engine-Version.

   Wenn Sie eine Amazon DocumentDB-Cluster-Parametergruppe als Teil der Upgrade-Anforderung zuordnen, startet das direkte Upgrade der Hauptversion den Cluster automatisch neu, um die neue Parametergruppe anzuwenden.
2. Stellen Sie sicher, dass Sie die Voraussetzungen für ein direktes Upgrade der Hauptversion erfüllt haben, wie im Abschnitt Voraussetzungen und Einschränkungen beschrieben.
3. Erstellen Sie einen manuellen Snapshot.

   Der Upgrade-Prozess erstellt während des Upgrades einen Snapshot Ihres Datenbank-Clusters. Es wird dringend empfohlen, vor dem Upgrade einen eigenen manuellen Snapshot zu erstellen. Siehe Erstellen eines manuellen Cluster-Snapshots.

> **Note**
>
> Der durch den Upgrade-Prozess erstellte automatische Snapshot wird nicht automatisch gelöscht, nachdem das direkte Upgrade der Hauptversion abgeschlossen ist. Für diesen Snapshot fallen keine Gebühren an, solange er innerhalb des Aufbewahrungszeitraums liegt. Sie können diesen Snapshot löschen, sobald Sie ein erfolgreiches Upgrade Ihres Clusters verifiziert haben.

Der Snapshot heißt `preupgrade-<name>-<version>-<timestamp>`.



4. Überprüfen Sie, ob Sie bereits ein direktes Hauptversions-Upgrade Ihres Clusters geplant haben.

   Wenn Sie den Cluster geändert und ausgewählt haben, um ihn im nächsten Wartungsfenster anzuwenden, ist der Zeitplan für das direkte Upgrade der Hauptversion in der Konsole nicht sichtbar, aber Sie können ihn in der CLI anzeigen. Sie können den folgenden Befehl ausführen, um zu überprüfen, ob ein direktes Hauptversions-Upgrade bereits geplant ist:

   ```
   aws docdb describe-db-cluster \
   --region $REGION \
   --db-cluster-identifier $CLUSTER_NAME

   "PendingModifiedValues": {
       "EngineVersion": "5.0.0"
   },
   ```

5. Führen Sie mehrere Testläufe durch, indem Sie den Volume-Klon in niedrigeren Umgebungen verwenden, um den Cluster nach dem In-Place-Hauptversions-Upgrade auf alle Ausführungspläne und Funktionsunterschiede zu testen. Wir empfehlen, mit derselben Anzahl und Größe von Instances zu klonen, um eine bessere Schätzung der Laufzeit des In-Place-Hauptversions-Upgrades zu erhalten. Weitere Informationen finden Sie unter Klonen eines Volumes für einen Amazon DocumentDB-Cluster.

6. Wenn der vorherige Schritt erfolgreich ist, fahren Sie mit dem direkten Upgrade der Hauptversion auf dem Produktions-Cluster fort.

Während eines direkten Hauptversions-Upgrades

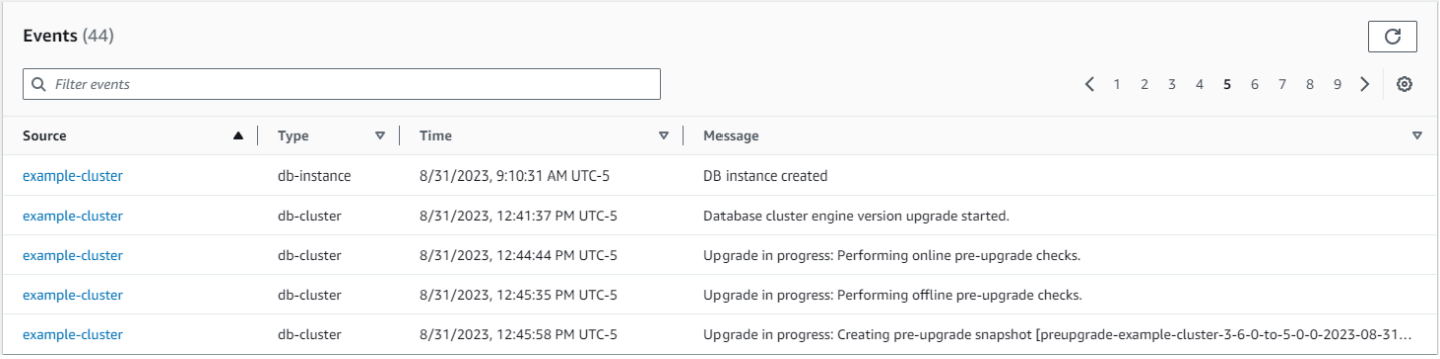
Sie können den Fortschritt Ihres direkten Hauptversions-Upgrades überwachen, indem Sie Cluster-Wartungsereignisse abonnieren. Wenn das Upgrade abgeschlossen ist, erhalten Sie das Ereignis „Datenbank-Cluster-Hauptversion wurde aktualisiert“. Diese und andere Ereignisse, die während des Upgrades auftreten, werden im Abschnitt „Ereignisse und Tags“ der Cluster-Detailseite in der Amazon DocumentDB-Konsole angezeigt. Der Clusterstatus ändert sich dann von „Upgrade“ auf „verfügbar“.

Von CLI aus können Sie ausführen, `aws docdb create-event-subscription` um Ereignisse zu erstellen und den Fortschritt `aws docdb describe-events` zu überwachen. Sie können auch Ereignisbenachrichtigungen für die oben genannten Ereignisse an Amazon SNS als Ziel einrichten, das per E-Mail, Push-Nachrichten und anderen Methoden benachrichtigt werden soll. Weitere Informationen finden Sie unter Abonnieren von Amazon DocumentDB DocumentDB-Ereignisabonnements.

Das direkte Upgrade der Hauptversion generiert während des Upgrades die folgenden Ereignisse:

- Upgrade läuft: Erstellen eines Snapshots vor dem Upgrade [preupgrade-<cluster-name>-<timestamp>]
- Upgrade läuft: Klonen des Volumes.
- Upgrade läuft: Writer wird aktualisiert.
- Upgrade läuft: Aktualisieren von Lesern.
- Die Hauptversion des Datenbank-Clusters wurde aktualisiert.

Ereignisse sind auch in der Konsole auf der Seite Ereignisse sichtbar:



In der können AWS CLI Sie den Fortschritt mit den folgenden Befehlen verfolgen:

```
aws docdb describe-events --source-identifier $CLUSTER_NAME --source-type db-cluster
{
    "Events": [
        {
            "SourceIdentifier": "mycluster",
            "SourceType": "db-cluster",
            "Message": "Database cluster engine version upgrade started.",
            "EventCategories": [
                "maintenance"
            ],
            "Date": "2023-07-11T23:20:32.444000+00:00",
            "SourceArn": "arn:aws:rds:us-east-1:xxxx:cluster:mycluster"
        }
    ]
}
```

Nach einem direkten Upgrade der Hauptversion

Fügen Sie für Amazon DocumentDB 3.6 dem Cluster ein Tag hinzu, um zu unterscheiden, dass der Cluster von Amazon DocumentDB 3.6 auf Amazon DocumentDB 5.0 aktualisiert wurde, im Gegensatz zu einem neu erstellten Amazon DocumentDB-5.0-Cluster. Weitere Informationen finden Sie im Abschnitt zu den Unterschieden zwischen einem aktualisierten Amazon DocumentDB-5.0-Cluster und einem neuen Amazon DocumentDB-5.0-Cluster.

Erstellen Sie einen manuellen Snapshot, nachdem das direkte Upgrade der Hauptversion abgeschlossen ist, falls Sie die Wiederherstellung in den Status nach dem Upgrade durchführen müssen. Der automatische Snapshot-Prozess wird fortgesetzt, sobald das direkte Upgrade der Hauptversion abgeschlossen ist. Für den manuellen Snapshot fallen keine Gebühren an, solange er innerhalb des Aufbewahrungszeitraums liegt.

Um die neuen Funktionen von Amazon DocumentDB 5.0 zu verwenden, z. B. die clientseitige Verschlüsselung auf Feldebene, empfehlen wir, Ihre Treiberversion auf die MongoDB-5.0-API-Version zu aktualisieren. Weitere Informationen finden Sie unter Was ist neu in Amazon DocumentDB 5.0 für eine Liste der Funktionen von Amazon DocumentDB 5.0.

> **Note**
>
> Beim Aufwärmen des Puffercache kommt es nach dem Upgrade zu einem vorübergehenden Leistungsabfall des Amazon DocumentDB-Clusters.

Testen Sie den aktualisierten Amazon DocumentDB-5.0-Cluster vollständig, um sicherzustellen, dass alles wie erwartet funktioniert.

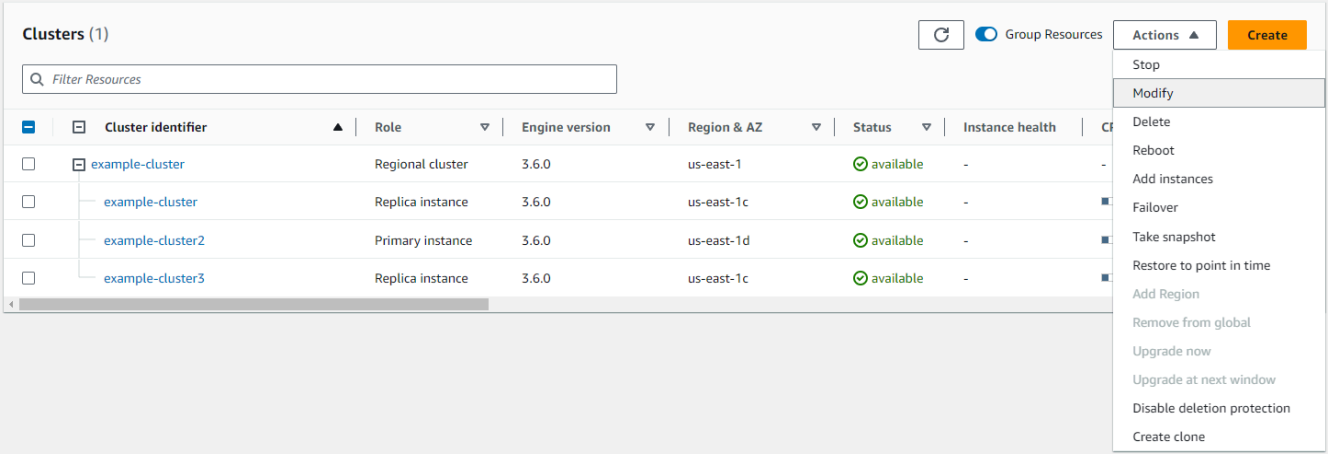
> **Note**
>
> Nach dem Ausführen einer direkten MVU auf einem Amazon DocumentDB-Cluster mit aktivierten Änderungsstreams werden die vorherigen Änderungsstream-Ereignisse beibehalten und können mit `resumeToken` oder fortgesetzt werden`startAtOperationTime`. Wie bei neu erstellten Amazon DocumentDB-Clustern werden auch Änderungsstream-Ereignisprotokolle gelöscht, die älter als`change_stream_log_retention_duration` sind, wenn die Protokollgröße größer als 51.200 MB ist.

Durchführen eines direkten Hauptversions-Upgrades

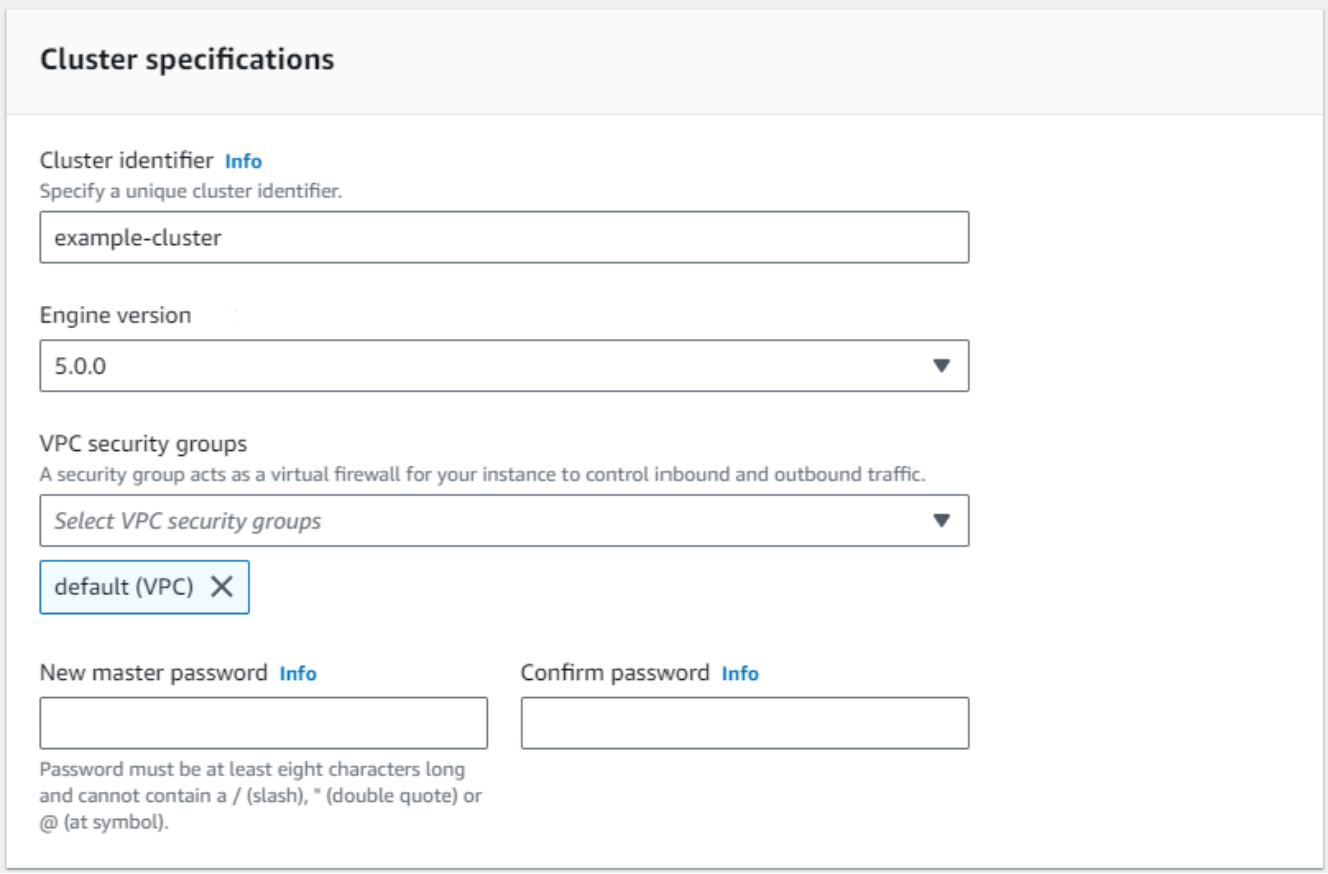
Using the AWS Management Console

So führen Sie ein direktes Hauptversions-Upgrade mit der durch AWS Management Console:

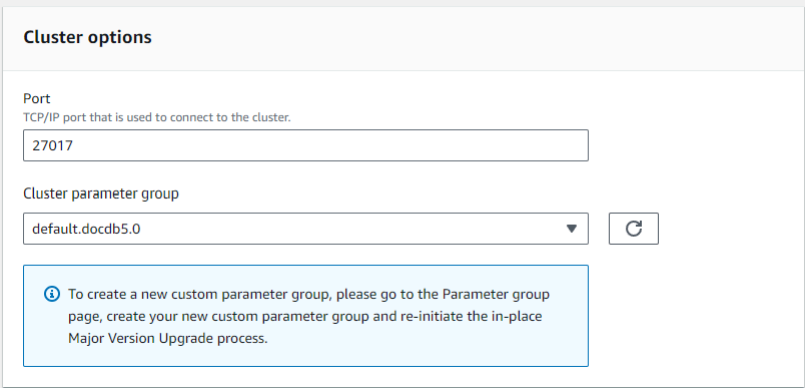
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Wählen Sie in der Tabelle Cluster den Quell-Cluster aus, klicken Sie auf Aktionen und dann auf Ändern.



3. Wählen Sie im Dialogfeld Cluster ändern im Abschnitt Cluster-Spezifikationen die gewünschte Datenbankversion (5.0) aus dem Dropdown-Menü Engine-Version aus.

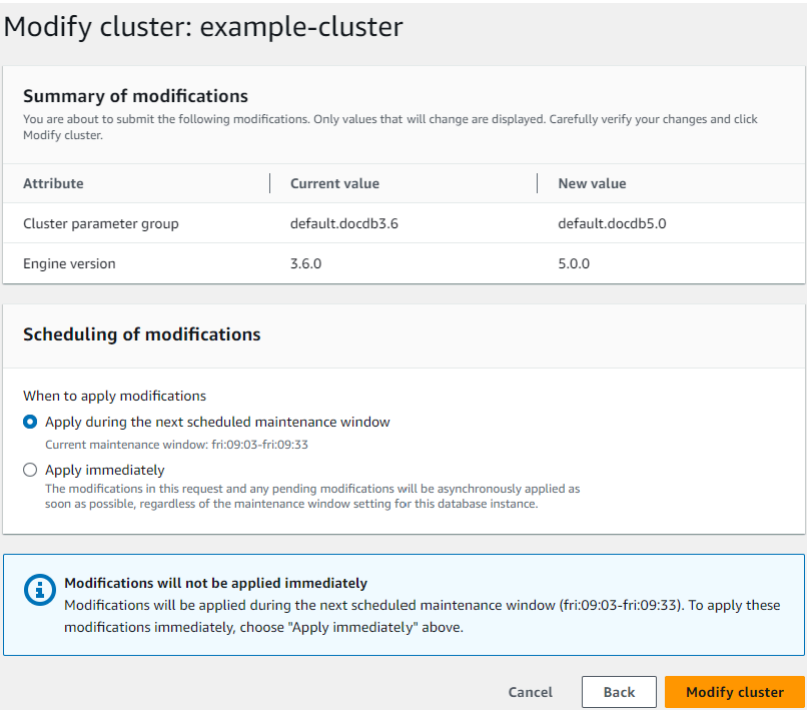


4. Wählen Sie im Abschnitt Cluster-Optionen die entsprechende Cluster-Parametergruppe (default.docdb5.0) oder eine benutzerdefinierte Parametergruppe aus.

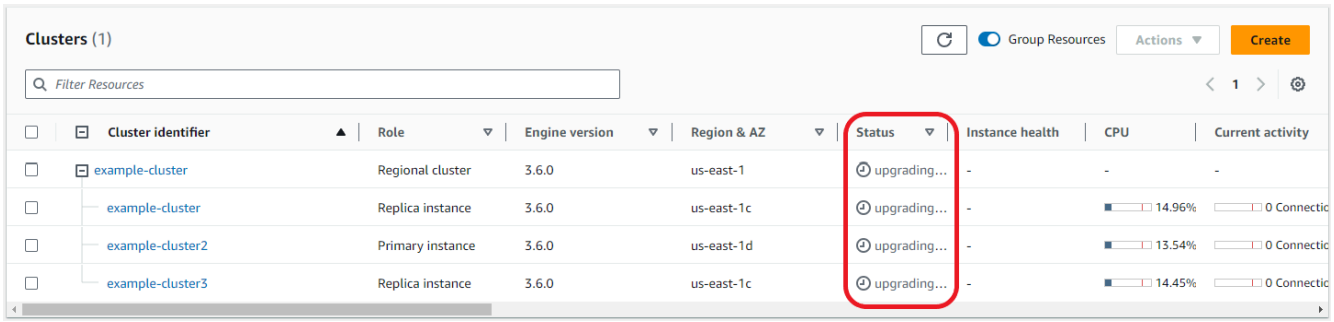


5. Wenn Sie fertig sind, scrollen Sie nach unten und wählen Sie Weiter aus.
6. Wählen Sie im Abschnitt Planen von Änderungen Ihren bevorzugten Planungsplan aus: Sofort anwenden oder im nächsten Wartungsfenster anwenden.

   Wählen Sie Cluster ändern aus.



7. Notieren Sie sich in der Tabelle Cluster den Status Ihres Clusters, während er aktualisiert wird:



Using the AWS CLI

Verwenden Sie die `modify-db-cluster` API mit der gewünschten Engine-Version und gesetztem `allow-major-version-upgrade` Flag:

```
aws docdb modify-db-cluster \
 --db-cluster-identifier $CLUSTER_NAME \
 --allow-major-version-upgrade \
 --engine-version 5.0 \
 --apply-immediately \
 --cluster-parameter-group $PARAMETER_GROUP \
 --region $REGION
```

Fehlerbehebung bei einem direkten Hauptversions-Upgrade

Im Falle eines Fehlers versucht das direkte Upgrade der Hauptversion, ein Rollback des Upgrades durchzuführen, um den letzten Betriebsstatus des Clusters zu übernehmen, bevor das Upgrade gestartet wurde. Ein erfolgreiches Rollback generiert ein Ereignis: „Datenbank-Cluster befindet sich in einem Zustand, der nicht aktualisiert werden kann: DocumentDB-Cluster befindet sich in einem Zustand, in dem das Hauptversions-Upgrade nicht erfolgreich abgeschlossen werden kann.“ An diesem Punkt sollten Sie sich an das AWS Support-Team wenden, um Probleme zu beheben und das Versionsupgrade erneut zu versuchen. Sie können Ihren Workload weiterhin wie zuvor verwenden. In anderen seltenen Szenarien, in denen das Upgrade länger als erwartet dauert, wenden Sie sich bitte an das AWS Support-Team, um Unterstützung zu erhalten.

Unterschiede zwischen aktualisierten Clustern von Amazon DocumentDB 3.6 auf 5.0 und neuen Clustern von Amazon DocumentDB 5.0

- Vergleiche von Unterdokumenten für mehrere numerische Datentypen:
  - Wenn der Cluster von Amazon DocumentDB 3.6 migriert wird, erbt er das Vergleichsverhalten des Amazon DocumentDB-3.6-Unterdokuments. Der funktionale Unterschied ist auf numerische Typen (wie Long, Double, Decimal128) in einem Unterdokument beschränkt. Beispielsweise `{a: {b: {NumberLong(1)}}` ist `{a: {b: 1}}` in Amazon DocumentDB 3.6 nicht gleich, während sie in Amazon DocumentDB 4.0 und danach als gleich verglichen werden.
  - Dieses Vergleichsverhalten für Unterdokumente ist nur in Amazon DocumentDB 3.6 und in Amazon DocumentDB-5.0-Clustern vorhanden, die mit einem direkten Upgrade der Hauptversion von Version 3.6 aktualisiert wurden. Dies gilt nicht für neu erstellte Amazon DocumentDB-5.0-Cluster.
- Ein direktes Upgrade der Hauptversion behält die ursprünglichen Indizes auf dem aktualisierten Cluster bei. Wir empfehlen, Ihre Indizes neu zu erstellen, um Amazon DocumentDB-5.0-spezifische Leistungsverbesserungen voll auszunutzen (z. B. Garbage Collection). Das Neuerstellen eines Indexes kann jedoch zusätzliche E/A und Zeit beinhalten. Weitere Informationen finden Sie unter Verwalten von Amazon DocumentDB-Indizes.

> **Note**
>
> Eine Liste der funktionalen Unterschiede zwischen Amazon DocumentDB 3.6/4.0 und Amazon DocumentDB 5.0 finden Sie unter MongoDB-Kompatibilität.

Sicherheit in Amazon DocumentDB

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die eingerichtet wurde, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Diese Dokumentation zeigt Ihnen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von Amazon DocumentDB einsetzen können. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Um mehr über die Compliance-Programme zu erfahren, die für Amazon DocumentDB gelten (mit MongoDB-Kompatibilität).AWS
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

> **Note**
>
> Dieses Kapitel gilt sowohl für instanzbasierte Cluster als auch für Elastic Clusters. Weitere Informationen finden Sie in den folgenden Themen.

Sie erfahren außerdem, wie Sie andereAWS -Services verwenden, um Ihre Amazon-DocumentDB-Ressourcen zu überwachen und zu schützen. Die folgenden Themen veranschaulichen, wie Sie Amazon DocumentDB so konfigurieren, dass Ihre Sicherheits- und Compliance-Ziele erreicht werden.

Themen

- Datenschutz in Amazon DocumentDB
- Identity and Access Management für Amazon DocumentDB
- Amazon-DocumentDB-Benutzer

- Datenbankzugriff mit rollenbasierter Zugriffskontrolle
- Logkolliert liert lierung in Amazon DocumentDB
- Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate
- Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate — GovCloud (US-West)
- Konformitätsprüfung in Amazon DocumentDB
- Ausfallsicherheit in Amazon DocumentDB
- Infrastruktursicherheit in Amazon DocumentDB
- Bewährte Sicherheitsmethoden für Amazon DocumentDB
- Amazon DocumentDB DocumentDB-Ereignisse prüfen

Datenschutz in Amazon DocumentDB

Das Modell der AWS geteilten gilt für den Datenschutz in . https://aws.amazon.com/compliance/shared-responsibility-model/ Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS-Modell der geteilten Verantwortung und in der DSGVO im AWS-Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.

- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon DocumentDB oder anderen AWS-Services über die Konsole, APIAWS CLI, oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Themen

Clientseitige Verschlüsselung auf Feldebene der Feldebene des Clients

Mit der clientseitigen Feldebenenverschlüsselung (FLE) von Amazon DocumentDB können Sie vertrauliche Daten in Ihren Client-Anwendungen verschlüsseln, bevor sie in einen Amazon DocumentDB-Cluster übertragen werden. Sensible Daten bleiben verschlüsselt, wenn sie in einem Cluster gespeichert und verarbeitet werden, und werden beim Abrufen in der Client-Anwendung entschlüsselt.

Themen

Erste Schritte

Die Erstkonfiguration von clientseitigem FLE in Amazon DocumentDB ist ein vierstufiger Prozess, der die Erstellung eines Verschlüsselungsschlüssels, die Zuordnung einer Rolle zur Anwendung, die Konfiguration der Anwendung und die Definition des CRUD-Betriebs mit Verschlüsselungsoptionen umfasst.

Themen

Schritt 1: Verschlüsseln des Verschlüsseln des Verschlüsseln des Verschlüsseln des

Erstellen Sie mithilfe einen symmetrischen SchlüsselAWS Key Management Service, der zum Verschlüsseln und Entschlüsseln des sensiblen Datenfeldes verwendet wird, und geben Sie ihm die erforderlichen IAM-Nutzungsberechtigungen. AWS KMSspeichert den Kundenschlüssel (CK), der zur Verschlüsselung von Datenschlüsseln (DKs) verwendet wird. Wir empfehlen, den Kundenschlüssel in KMS zu speichern, um Ihre Sicherheitslage zu verbessern. Der Datenschlüssel ist der Sekundärschlüssel, der in einer Amazon DocumentDB-Sammlung gespeichert wird und erforderlich ist, um sensible Felder zu verschlüsseln, bevor das Dokument in Amazon DocumentDB gespeichert wird. Der Kundenschlüssel verschlüsselt den Datenschlüssel, der wiederum Ihre Daten ver- und entschlüsselt. Wenn Sie einen globalen Cluster verwenden, können Sie einen Schlüssel mit mehreren Regionen erstellen, der von verschiedenen Servicerollen in verschiedenen Regionen verwendet werden kann.

Weitere Informationen zumAWS Key Management Service, einschließlich der Erstellung eines Schlüssels, finden Sie im AWSKey Management Service Developer Guide.

Schritt 2: Zuordnen einer Rolle zu der Anwendung des Antrags des Antrags des Antrags

Erstellen einer IAM-Richtlinie mit den entsprechendenAWS KMS Berechtigungen des IAM-Schlüssels des IAM-Schlüssels Diese Richtlinie erlaubt es IAM-Identitäten, denen sie angefügt ist, den KMS-Schlüssel, denen sie angefügt ist, den KMS-Schlüssel, denen sie angefügt ist, den KMS-Schlüssel, denen sie angefügt ist. Ihre Anwendung übernimmt diese IAM-Rolle für die AuthentifizierungAWS KMS.

Die Richtlinie sollte in etwa so aussehen:

```
{ "Effect": "Allow",
"Action": ["kms:Decrypt", "kms:Encrypt"],
"Resource": "Customer Key ARN"
}
```

Schritt 3: Konfigurieren der Anwendung des Programms des Programmes des Programms

Inzwischen haben Sie einen Kundenschlüssel definiertAWS KMS und eine IAM-Rolle erstellt und ihr die richtigen IAM-Berechtigungen für den Zugriff auf den Kundenschlüssel zugewiesen. Importieren Sie die erforderlichen Pakete.

```
import boto3
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                                ClientEncryption)
```

```
# create a session object:
my_session = boto3.session.Session()

# get access_key and secret_key programmatically using get_frozen_credentials() method:
 current_credentials = my_session.get_credentials().get_frozen_credentials()
```

1. Geben Sie 'aws' als KMS-Anbietertyp an und geben Sie Ihre Kontoanmeldeinformationen ein, die im vorherigen Schritt abgerufen wurden.

   ```
   provider = "aws"
   kms_providers = {
       provider: {
           "accessKeyId": current_credentials.access_key,
           "secretAccessKey": current_credentials.secret_key
       }
   }
   ```

2. Geben Sie den Kundenschlüssel zum Verschlüsseln des Verschlüsseln des Verschlüsseln des Schlüssels des Schlüssels des Schlüssels des Schlüssels des Schlüssels

   ```
   customer_key = {
   ```

```
“region”: “AWS region of the customer_key”,
    “key”: “customer_key ARN”
}

key_vault_namespace = "encryption.dataKeys"

key_alt_name = 'TEST_DATA_KEY'
```

3. Konfigurieren Sie das MongoClient Objekt:

```
client = MongoClient(connection_string)

coll = client.test.coll
coll.drop()

client_encryption = ClientEncryption(
    kms_providers, # pass in the kms_providers variable from the previous step
    key_vault_namespace = key_vault_namespace,
    client,
    coll.codec_options
)
```

4. Generieren Sie Ihren Datenschlüssel:

```
data_key_id = client_encryption.create_data_key(provider,
    customer_key,
    key_alt_name = [key_alt_name])
```

5. Rufen Sie Ihren vorhandenen Datenschlüssel ab:

```
data_key = DataKey("aws",
    master_key = customer_key)
key_id = data_key["_id"]
data_key_id = client[key_vault_namespace].find_one({"_id": key_id})
```

Schritt 4: Definieren Sie eine CRUD-Operation

Definieren Sie den CRUD-Vorgang mit Verschlüsselungsoptionen.

1. Definieren Sie die Sammlung, um ein einzelnes Dokument zu schreiben/lesen/zu löschen:

```
coll = client.gameinfo.users
```

2. Explizite Verschlüsselung — Felder verschlüsseln und einfügen:

> **Note**
>
> Es muss genau eine der Optionen „key_id“ oder „key_alt_name“ angegeben werden.

```
encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_last_name = client_encryption.encrypt(
    "Doe",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_dob = client_encryption.encrypt(
    "1990-01-01",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Random,
    key_alt_name=data_key_id
)

coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name,
    "lastName": encrypted_last_name,
    "dateOfBirth":encrypted_dob,
    "Favorite_games":["Halo","Age of Empires 2","Medal of Honor"]
})
```

Beispiel: Konfigurationsdatei für die clientseitige Verschlüsselung auf Feldebene

Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

```
# import python packages:
import boto3
```

```
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                                ClientEncryption)

def main():

    # create a session object:
    my_session = boto3.session.Session()

    # get aws_region from session object:
    aws_region = my_session.region_name

    # get access_key and secret_key programmatically using get_frozen_credentials()
 method:
    current_credentials = my_session.get_credentials().get_frozen_credentials()
    provider = "aws"

    # define the kms_providers which is later used to create the Data Key:
    kms_providers = {
        provider: {
            "accessKeyId": current_credentials.access_key,
            "secretAccessKey": current_credentials.secret_key
        }
    }

    # enter the kms key ARN. Replace the example ARN value.
    kms_arn = "arn:aws:kms:us-east-1:123456789:key/abcd-efgh-ijkl-mnop"
    customer_key = {
        "region": aws_region,
        "key":kms_arn
    }

    # secrets manager is used to strore and retrieve user credentials for connecting to
 an Amazon DocumentDB cluster.
    # retrieve the secret using the secret name. Replace the example secret key.
    secret_name = "/dev/secretKey"
    docdb_credentials = json.loads(my_session.client(service_name = 'secretsmanager',
 region_name = "us-east-1").get_secret_value(SecretId = secret_name)['SecretString'])

    connection_params = '/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
```

```
    conn_str = 'mongodb://' + docdb_credentials["username"] + ':' +
docdb_credentials["password"] + '@' + docdb_credentials["host"] + ':' +
str(docdb_credentials["port"]) + connection_params
    client = MongoClient(conn_str)

    coll = client.test.coll
    coll.drop()

    # store the encryption data keys in a key vault collection (having naming
convention as db.collection):
    key_vault_namespace = "encryption.dataKeys"
    key_vault_db_name, key_vault_coll_name = key_vault_namespace.split(".", 1)

    # set up the key vault (key_vault_namespace) for this example:
    key_vault = client[key_vault_db_name][key_vault_coll_name]
    key_vault.drop()
    key_vault.create_index("keyAltNames", unique=True)

    client_encryption = ClientEncryption(
        kms_providers,
        key_vault_namespace,
        client,
        coll.codec_options)

    # create a new data key for the encrypted field:
    data_key_id = client_encryption.create_data_key(provider, master_key=customer_key,
key_alt_names=["some_key_alt_name"], key_material = None)

    # explicitly encrypt a field:
    encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_id=data_key_id
    )
    coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name
    })
    doc = coll.find_one()
    print('Encrypted document: %s' % (doc,))

    # explicitly decrypt the field:
    doc["encryptedField"] = client_encryption.decrypt(doc["encryptedField"])
    print('Decrypted document: %s' % (doc,))
```

```
    # cleanup resources:
    client_encryption.close()
    client.close()

    if __name__ == "__main__":
        main()
```

Abfragen in der clientseitigen FLE des clientseitigen FLE des Clients

Amazon DocumentDB unterstützt Punktegleichheitsabfragen mit clientseitigem FLE. Ungleichheits- und Vergleichsabfragen können zu ungenauen Ergebnissen führen. Lese- und Schreibvorgänge können ein unerwartetes oder falsches Verhalten aufweisen, verglichen mit der Ausführung derselben Operation für den entschlüsselten Wert.

Um beispielsweise Filter für Dokumente abzufragen, bei denen der Gamerscore größer als 500 ist:

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
})
```

Der Client verwendet eine explizite Verschlüsselungsmethode, um den Abfragewert zu verschlüsseln:

```
encrypted_gamerscore_filter = client_encryption.encrypt(
    500,
        Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
        key_alt_name=data_key_id
        )

db.users.find( {
    "gamerscore" : { $gt : encrypted_gamerscore_filter }
} )
```

Beim Suchvorgang vergleicht Amazon DocumentDB den verschlüsselten Wert von 500 mit den verschlüsselten Feldwerten, die in jedem Dokument gespeichert sind. Dabei wird der Wert größer als die Ungleichheit geprüft. Die Ungleichheitsprüfung im Suchvorgang kann zu einem anderen Ergebnis führen, wenn sie mit entschlüsselten Daten und Werten ausgeführt wird, obwohl die Operation erfolgreich Ergebnisse generiert.

Einschränkungen

Die folgenden Einschränkungen gelten für die clientseitige Verschlüsselung auf Feldebene von Amazon DocumentDB:

- Amazon DocumentDB unterstützt nur Abfragen zur Punktegleichheit. Ungleichheits- und Vergleichsabfragen können zu ungenauen Ergebnissen führen. Lese- und Schreibvorgänge können ein unerwartetes oder falsches Verhalten aufweisen, verglichen mit der Ausführung derselben Operation für den entschlüsselten Wert. Um Filter für Dokumente abzufragen, bei denen der Gamerscore größer als 500 ist.

  ```
  db.users.find( {
      "gamerscore" : { $gt : 500 }
      })
  ```

  Der Client verwendet eine explizite Verschlüsselungsmethode, um den Abfragewert zu verschlüsseln.

  ```
  encrypted_gamerscore_filter = client_encryption.encrypt(
      500,
      Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
      key_alt_name=data_key_id
  )

  db.users.find({
      "gamerscore" : { $gt : encrypted_gamerscore_filter }
  })
  ```

  Beim Suchvorgang vergleicht Amazon DocumentDB den verschlüsselten Wert von 500 mit den verschlüsselten Feldwerten, die in jedem Dokument gespeichert sind. Dabei wird der Wert größer als die Ungleichheit geprüft. Die Ungleichheitsprüfung im Suchvorgang kann zu einem anderen Ergebnis führen, wenn sie mit entschlüsselten Daten und Werten ausgeführt wird, obwohl die Operation erfolgreich Ergebnisse generiert.
- Amazon DocumentDB unterstützt keine explizite clientseitige FLE aus der Mongo Shell. Die Funktion funktioniert jedoch mit jedem unserer unterstützten Treiber.

Verschlüsselung ruhender Amazon DocumentDB DocumentDB-Daten

> **Note**
>
> AWS KMS ersetzt den Begriff Kundenhauptschlüssel (CMK) durch AWS KMS key und KMS-Schlüssel. Das Konzept hat sich nicht geändert. Um abwärtsinkompatible Änderungen zu vermeiden, werden von AWS KMS einige Varianten dieses Begriffs beibehalten.

Sie verschlüsseln ruhende Daten in Ihrem Amazon DocumentDB-Cluster, indem Sie bei der Erstellung Ihres Clusters die Speicherverschlüsselungsoption angeben. Die Speicher-Verschlüsselung ist Cluster-weit aktiviert und wird auf alle Instances angewendet, einschließlich der primären Instance und alle Replikate. Sie wird auch auf das Speichervolumen Ihres Clusters, auf Daten, Indizes, Protokolle, automatisierte Sicherungen und Snapshots angewendet.

Amazon DocumentDB verwendet den 256-Bit Advanced Encryption Standard (AES-256), um Ihre Daten mit Verschlüsselungsschlüsseln zu verschlüsseln.AWS Key Management ServiceAWS KMS Wenn Sie einen Amazon DocumentDB-Cluster mit aktivierter Verschlüsselung im Ruhezustand verwenden, müssen Sie Ihre Anwendungslogik oder Ihre Client-Verbindung nicht ändern. Amazon DocumentDB übernimmt die Verschlüsselung und Entschlüsselung Ihrer Daten auf transparente Art und Weise und mit minimaler Auswirkung auf die Leistung.

Amazon DocumentDB lässt sich in eine als Envelope-Verschlüsselung bekannte Methode integrierenAWS KMS und verwendet diese, um Ihre Daten zu schützen. Wenn ein Amazon DocumentDB-Cluster mit einem verschlüsselt wirdAWS KMS, fordert Amazon DocumentDB Sie auf, Ihren KMS-SchlüsselAWS KMS zu verwenden, um einen Chiffretext-Datenschlüssel zur Verschlüsselung des Speichervolumes zu generieren. Der Chiffretext-Datenschlüssel wird mit dem von Ihnen definierten KMS-Schlüssel verschlüsselt und zusammen mit den verschlüsselten Daten und Speichermetadaten gespeichert. Wenn Amazon DocumentDB auf Ihre verschlüsselten Daten zugreifen muss, fordertAWS KMS es die Entschlüsselung des Chiffretext-Datenschlüssels mithilfe Ihres KMS-Schlüssels an und speichert den Klartext-Datenschlüssel im Speicher, um Daten auf dem Speichervolume effizient zu ver- und entschlüsseln.

Die Speicherverschlüsselungsfunktion in Amazon DocumentDB ist für alle unterstützten Instanzgrößen und in allen Ländern verfügbar, inAWS-Regionen denen Amazon DocumentDB verfügbar ist.

Verschlüsselung im Ruhezustand für einen Amazon DocumentDB-Cluster aktivieren

Sie können die Verschlüsselung im Ruhezustand auf einem Amazon DocumentDB-Cluster aktivieren oder deaktivieren, wenn der Cluster entweder mit demAWS Management Console oder derAWS Command Line Interface (AWS CLI) bereitgestellt wird. Cluster, die Sie mit der Konsole erstellen, haben standardmäßig die Verschlüsselung im Ruhezustand aktiviert. Cluster, die Sie mit der AWS CLI erstellen, haben standardmäßig die Verschlüsselung im Ruhezustand deaktiviert. Daher müssen Sie die Verschlüsselung im Ruhezustand explizit mit dem `--storage-encrypted`-Parameter aktivieren. In beiden Fällen können Sie nach dem Erstellen des Clusters die Option „Verschlüsselung im Ruhezustand“ nicht ändern.

Amazon DocumentDB verwendet,AWS KMS um Verschlüsselungsschlüssel abzurufen und zu verwalten und die Richtlinien zu definieren, die steuern, wie diese Schlüssel verwendet werden können. Wenn Sie keineAWS KMS Schlüssel-ID angeben, verwendet Amazon DocumentDB den standardmäßigen KMS-Schlüssel fürAWS verwaltete Dienste. Amazon DocumentDB erstellt für jedenAWS-Region in Ihrem einen eigenen KMS-SchlüsselAWS-Konto. Weitere Informationen finden Sie unter AWS Key Management Service-Konzepte.

Informationen zum Erstellen Ihres eigenen KMS-Schlüssels finden Sie unter Erste Schritte im AWS Key Management ServiceEntwicklerhandbuch.

> **Important**
>
> Sie müssen einen symmetrischen KMS-Verschlüsselungsschlüssel verwenden, um Ihren Cluster zu verschlüsseln, da Amazon DocumentDB nur KMS-Verschlüsselungsschlüssel unterstützt. Verwenden Sie keinen asymmetrischen KMS-Schlüssel, um zu versuchen, die Daten in Ihren Amazon DocumentDB-Cluster zu verschlüsseln. Weitere Informationen finden Sie unter Asymmetrische SchlüsselAWS KMS imAWS Key Management Service Entwicklerhandbuch.

Wenn Amazon DocumentDB keinen Zugriff auf den Verschlüsselungsschlüssel für einen Cluster hat - zum Beispiel, wenn der Zugriff auf einen Schlüssel widerrufen wird - geht der verschlüsselte Cluster in einen Endzustand über. In diesem Fall können Sie den Cluster nur aus einer Sicherung wiederherstellen. Für Amazon DocumentDB sind Backups immer für einen Tag aktiviert.

Wenn Sie den Schlüssel für einen verschlüsselten Amazon DocumentDB-Cluster deaktivieren, verlieren Sie außerdem irgendwann den Lese- und Schreibzugriff auf diesen Cluster. Wenn Amazon

DocumentDB auf einen Cluster trifft, der durch einen Schlüssel verschlüsselt ist, auf den es keinen Zugriff hat, versetzt es den Cluster in einen Endzustand. In diesem Fall ist der Cluster nicht länger verfügbar und der aktuelle Zustand der Datenbank kann nicht mehr wiederhergestellt werden. Um den Cluster wiederherzustellen, müssen Sie den Zugriff auf den Verschlüsselungsschlüssel für Amazon DocumentDB erneut aktivieren und den Cluster anschließend aus einer Sicherungsdatei wiederherstellen.

> **Important**
>
> Sie können den KMS-Schlüssel für einen verschlüsselten KMS-Schlüssel nicht mehr ändern. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel bestimmen, bevor Sie Ihren verschlüsselten Cluster erstellen.

Using the AWS Management Console

Geben Sie beim Erstellen eines Clusters die Option „Verschlüsselung im Ruhezustand“ an. Die Verschlüsselung im Ruhezustand ist standardmäßig aktiviert, wenn Sie einen Cluster mit der AWS Management Console erstellen. Nachdem der Cluster erstellt wurde, kann er nicht mehr geändert werden.

So legen Sie beim Erstellen eines Clusters die Option „Verschlüsselung im Ruhezustand“ fest

1. Erstellen Sie einen Amazon DocumentDB-Cluster, wie im Abschnitt Erste Schritte beschrieben. Wählen Sie jedoch in Schritt 6 nicht Create Cluster (Cluster erstellen) aus.
2. Wählen Sie unterhalb des Abschnitts Authentication (Authentifizierung) die Option Show advanced settings (Erweiterte Einstellungen anzeigen) aus.
3. Scrollen Sie nach unten bis zumncryption-at-rest Abschnitt E.
4. Wählen Sie die Option aus, die Sie für die Verschlüsselung im Ruhezustand wünschen. Egal, welche Option Sie wählen, nach dem Erstellen des Clusters können Sie sie nicht mehr ändern.
   - Wenn Sie Daten im Ruhezustand in diesem Cluster verschlüsseln möchten, wählen Sie Enable encryption (Verschlüsselung aktivieren) aus.
   - Wenn Sie Daten im Ruhezustand in diesem Cluster nicht verschlüsseln möchten, wählen Sie Disable encryption (Verschlüsselung deaktivieren) aus.
5. Wählen Sie den gewünschten Hauptschlüssel. Amazon DocumentDB verwendet dieAWS Key Management Service (AWS KMS), um Verschlüsselungsschlüssel abzurufen und zu

verwalten und um die Richtlinien zu definieren, die steuern, wie diese Schlüssel verwendet werden können. Wenn Sie keineAWS KMS Schlüssel-ID angeben, verwendet Amazon DocumentDB den standardmäßigen KMS-Schlüssel fürAWS verwaltete Dienste. Weitere Informationen finden Sie unter AWS Key Management Service-Konzepte.

> **Note**
>
> Nachdem Sie einen verschlüsselten Cluster erstellt haben, können Sie den KMS-Schlüssel für diesen Cluster nicht mehr ändern. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel bestimmen, bevor Sie Ihren verschlüsselten Cluster erstellen.

6. Füllen Sie die anderen Abschnitte nach Bedarf aus und erstellen Sie Ihren Cluster.

Using the AWS CLI

Um einen Amazon DocumentDB-Cluster mit dem zu verschlüsselnAWS CLI, müssen Sie die`--storage-encrypted` Option bei der Erstellung des Clusters angeben. Amazon DocumentDB-Cluster, die mit dem erstellt wurden, aktivieren standardmäßigAWS CLI keine Speicherverschlüsselung.

Im folgenden Beispiel wird ein Amazon DocumentDB-Cluster erstellt, in dem die Speicherverschlüsselung aktiviert ist.

Example

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --port 27017 \
      --engine docdb \
      --master-username yourMasterUsername \
      --master-user-password yourMasterPassword \
      --storage-encrypted
```

Für Windows:

```
aws docdb create-db-cluster ^
```

```
      --db-cluster-identifier sample-cluster ^
      --port 27017 ^
      --engine docdb ^
      --master-username yourMasterUsername ^
      --master-user-password yourMasterPassword ^
      --storage-encrypted
```

Wenn Sie einen verschlüsselten Amazon DocumentDB-Cluster erstellen, können Sie wie im folgenden Beispiel eineAWS KMS Schlüssel-ID angeben.

Example

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --port 27017 \
      --engine docdb \
      --master-username yourMasterUsername \
      --master-user-password yourMasterPassword \
      --storage-encrypted \
      --kms-key-id key-arn-or-alias
```

Für Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --port 27017 ^
      --engine docdb ^
      --master-username yourMasterUsername ^
      --master-user-password yourMasterPassword ^
      --storage-encrypted ^
      --kms-key-id key-arn-or-alias
```

> **Note**
>
> Nachdem Sie einen verschlüsselten Cluster erstellt haben, können Sie den KMS-Schlüssel für diesen Cluster nicht mehr ändern. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel bestimmen, bevor Sie Ihren verschlüsselten Cluster erstellen.

Einschränkungen für verschlüsselte Amazon DocumentDB-Cluster

Folgende Einschränkungen bestehen für Amazon DocumentDB-verschlüsselte Amazon DocumentDB-Cluster.

- Sie können die Verschlüsselung im Ruhezustand für einen Amazon DocumentDB-Cluster nur zum Zeitpunkt seiner Erstellung aktivieren oder deaktivieren, nicht nachdem der Cluster erstellt wurde. Sie können jedoch eine verschlüsselte Kopie eines unverschlüsselten Clusters erstellen, indem Sie einen Snapshot des unverschlüsselten Clusters erstellen und den unverschlüsselten Snapshot dann als neuen Cluster wiederherstellen, während Sie die Option Verschlüsselung im Ruhezustand angeben.

  Weitere Informationen finden Sie unter den folgenden Themen:

  - Erstellen eines manuellen Cluster-Snapshots
  - Wiederherstellen aus einem Cluster-Snapshot
  - Kopieren von Amazon DocumentDB-Cluster-Snapshots
- Amazon DocumentDB-Cluster mit aktivierter Speicherverschlüsselung können nicht geändert werden, um die Verschlüsselung zu deaktivieren.
- Alle Instances, automatisierten Backups, Snapshots und Indizes in einem Amazon DocumentDB-Cluster sind mit demselben KMS-Schlüssel verschlüsselt.

Datenverschlüsselung während der Übertragung

Sie können Transport Layer Security (TLS) verwenden, um die Verbindung zwischen Ihrer Anwendung und einem Amazon DocumentDB-Cluster zu verschlüsseln. Standardmäßig ist die Verschlüsselung bei der Übertragung für neu erstellte Amazon DocumentDB-Cluster aktiviert. Sie kann bei der Erstellung des Clusters oder zu einem späteren Zeitpunkt optional deaktiviert werden. Wenn die Verschlüsselung während der Übertragung aktiviert ist, sind sichere Verbindungen mit TLS erforderlich, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen zum Herstellen einer Verbindung zu Amazon DocumentDB über TLS finden Sie unter Programmgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB.

TLS-Einstellungen für Amazon DocumentDB-Cluster verwalten

Die Verschlüsselung während der Übertragung für einen Amazon DocumentDB-Cluster wird über den TLS-Parameter in einer Cluster-Parametergruppe verwaltet. Sie können Ihre Amazon DocumentDB-Cluster-TLS-Einstellungen mit dem AWS Management Console oder dem AWS Command Line

Interface (AWS CLI) verwalten. In den folgenden Abschnitten finden Sie weitere Informationen zum Überprüfen und Ändern Ihrer aktuellen TLS-Einstellungen.

Using the AWS Management Console

Gehen Sie wie folgt vor, um Verwaltungsaufgaben für die TLS-Verschlüsselung mithilfe der Konsole durchzuführen, z. B. Parametergruppen zu identifizieren, den TLS-Wert zu überprüfen und die erforderlichen Änderungen vorzunehmen.

> **Note**
>
> Sofern Sie beim Erstellen eines Clusters keine andere Angabe machen, wird Ihr Cluster mit der standardmäßigen Cluster-Parametergruppe erstellt. Die Parameter in der `default`-Cluster-Parametergruppe können nicht geändert werden (z. B. `tls` ist aktiviert/deaktiviert). Wenn Ihr Cluster also eine `default`-Cluster-Parametergruppe verwendet, müssen Sie den Cluster so ändern, dass er eine nicht standardmäßige Cluster-Parametergruppe verwendet. Zuerst müssen Sie möglicherweise eine benutzerdefinierte Cluster-Parametergruppe erstellen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen.

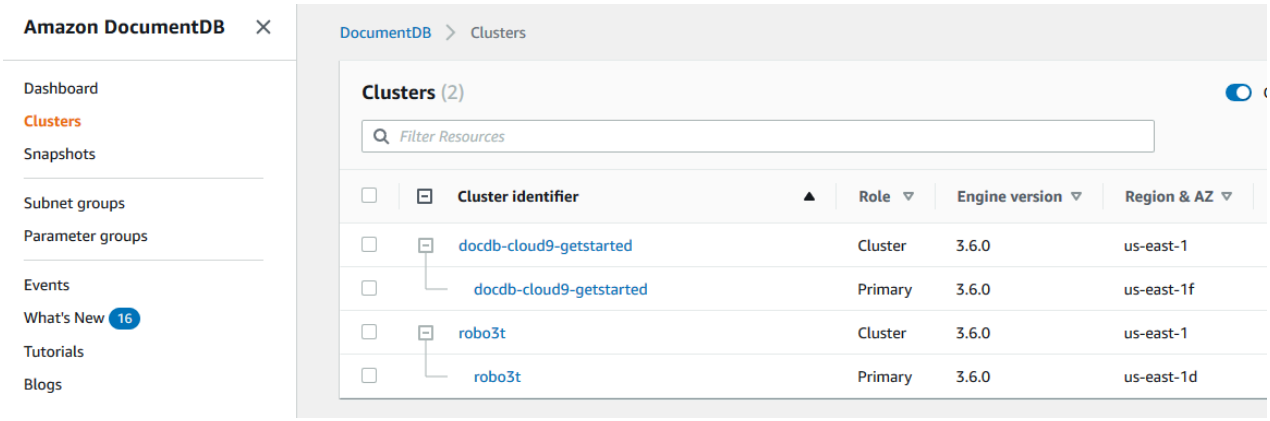
1. Bestimmen Sie, welche Cluster-Parametergruppe Ihr Cluster verwendet.

   a. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

   b. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

   c. Beachten Sie, dass im Navigationsfeld „Cluster“ in der Spalte Cluster Identifier sowohl Cluster als auch Instances angezeigt werden. Instances werden unter Clustern aufgeführt. Sehen Sie sich den Screenshot unten als Referenz an.



d. Wählen Sie den Cluster aus, an dem Sie interessiert sind.

e. Wählen Sie die Registerkarte Konfiguration und scrollen Sie bis zum Ende der Cluster-Details und suchen Sie die Cluster-Parametergruppe. Der Name der Cluster-Parametergruppe.

   Wenn der Name der Cluster-Parametergruppe `default` lautet (z. B. `default.docdb3.6`), müssen Sie eine benutzerdefinierte Cluster-Parametergruppe erstellen und diese zur Parametergruppe des Clusters machen, bevor Sie fortfahren. Weitere Informationen finden Sie hier:

   1. Amazon DocumentDB-Cluster-Parametergruppen erstellen— Wenn Sie keine benutzerdefinierte Cluster-Parametergruppe haben, die Sie verwenden können, erstellen Sie eine.
   2. Ändern eines Amazon DocumentDB-Clusters— Ändern Sie Ihren Cluster so, dass er die benutzerdefinierte Cluster-Parametergruppe verwendet.

2. Bestimmen Sie den aktuellen Wert des **`tls`**-Cluster-Parameters.

   a. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

   b. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   c. Wählen Sie aus der Liste der Cluster-Parametergruppen den Namen der von Ihnen gewünschten Cluster-Parametergruppe aus.

   d. Suchen Sie den Abschnitt Cluster-Parameter. Suchen Sie in der Liste der Cluster-Parameter die Zeile des `tls`-Cluster-Parameters. An dieser Stelle sind die folgenden vier Spalten wichtig:

- Cluster-Parametername — Der Name der Cluster-Parameter. Für die Verwaltung von TLS benötigen Sie den `tls`-Cluster-Parameter.
- Werte — Der aktuelle Wert jedes Cluster-Parameters.
- Zulässige Werte — Eine Liste von Werten, die auf einen Cluster-Parameter angewendet werden können.
- Typ anwenden — Entweder statisch oder dynamisch. Änderungen an statischen Cluster-Parametern können nur übernommen werden, wenn die Instances neu gestartet werden. Änderungen an dynamischen Cluster-Parametern können entweder sofort übernommen werden oder wenn die Instances neu gestartet werden.

3. Ändern Sie den Wert des **`tls`**-Cluster-Parameters.

   Wenn der Wert für `tls` nicht der benötigte Wert ist, ändern Sie ihn für diese Cluster-Parametergruppe. Um den Wert des `tls`-Cluster-Parameters zu ändern, fahren Sie nach dem vorherigen Abschnitt mit folgenden Schritten fort.

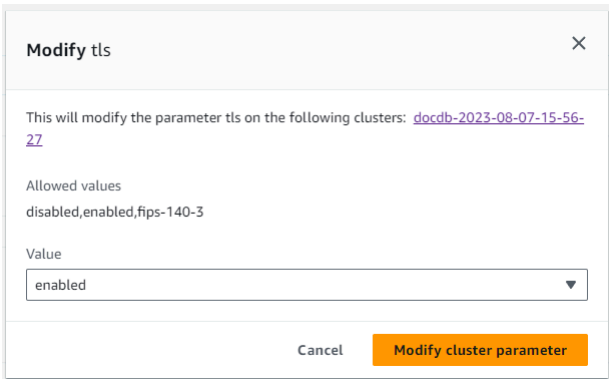
   a. Wählen Sie die Schaltfläche links neben dem Namen des Cluster-Parameters (`tls`).

   b. Wählen Sie Bearbeiten aus.

   c. Um den Wert von zu ändern`tls`, wählen Sie im `tls` Dialogfeld Ändern in der Dropdownliste den gewünschten Wert für den Cluster-Parameter aus.

      Gültige Werte für sind:

      - deaktiviert — Deaktiviert TLS
      - aktiviert — Aktiviert TLS
      - fips-140-3 — Aktiviert TLS mit FIPS. Der Cluster akzeptiert nur sichere Verbindungen gemäß den Anforderungen der Veröffentlichung 140-3 der Federal Information Processing Standards (FIPS). Dies wird erst ab Amazon DocumentDB 5.0-Clustern (Engine-Version 3.0.3727) in diesen Regionen unterstützt: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west



d. Wählen Sie Modify Cluster Parameter (Cluster-Parameter ändern). Die Änderung wird beim Neustart auf jede Cluster-Instance angewendet.

4. Starten Sie die Amazon DocumentDB DocumentDB-Instance neu.

   Starten Sie jede Instance des Clusters neu, sodass die Änderung für alle Instances im Cluster übernommen wird.

   a. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

   b. Wählen Sie im Navigationsbereich Instances aus.

   c. Um eine Instance anzugeben, die neu gestartet werden soll, suchen Sie die Instance in der Liste der Instances und wählen Sie die Schaltfläche links neben dem Namen aus.

   d. Wählen Sie Actions (Aktionen) und dann Reboot (Neustart) aus. Bestätigen Sie, dass Sie neu starten möchten, indem Sie auf Reboot (Neustart) klicken.

Using the AWS CLI

Gehen Sie wie folgt vor, um Verwaltungsaufgaben für die TLS-Verschlüsselung mithilfe von durchzuführenAWS CLI, z. B. Parametergruppen zu identifizieren, den TLS-Wert zu überprüfen und die erforderlichen Änderungen vorzunehmen.

> **Note**
>
> Sofern Sie beim Erstellen eines Clusters keine andere Angabe machen, wird Ihr Cluster mit der standardmäßigen Cluster-Parametergruppe erstellt. Die Parameter in der `default`-Cluster-Parametergruppe können nicht geändert werden (z. B. `tls` ist aktiviert/deaktiviert). Wenn Ihr Cluster also eine `default`-Cluster-Parametergruppe verwendet,

müssen Sie den Cluster so ändern, dass er eine nicht standardmäßige Cluster-Parametergruppe verwendet. Möglicherweise müssen Sie zuerst eine benutzerdefinierte Cluster-Parametergruppe erstellen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen.

1. Bestimmen Sie, welche Cluster-Parametergruppe Ihr Cluster verwendet.

   Verwenden Sie den `describe-db-clusters`-Befehl mit den folgenden Parametern:

   - **`--db-cluster-identifier`** – Erforderlich. Der Name des gewünschten Clusters.
   - **`--query`**— Fakultativ. Eine Abfrage, die die Ausgabe auf die gewünschten Felder begrenzt – in diesem Fall auf den Cluster-Namen und den Namen der Cluster-Parametergruppe.

   ```
   aws docdb describe-db-clusters \
          --db-cluster-identifier docdb-2019-05-07-13-57-08 \
          --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

   ```
   [
          [
              "docdb-2019-05-07-13-57-08",
              "custom3-6-param-grp"
          ]
   ]
   ```

   Wenn der Name der Cluster-Parametergruppe `default` lautet (z. B. `default.docdb3.6`), benötigen Sie eine benutzerdefinierte Cluster-Parametergruppe und müssen diese zur Parametergruppe des Clusters machen, bevor Sie fortfahren. Weitere Informationen finden Sie unter den folgenden Themen:

   1. Amazon DocumentDB-Cluster-Parametergruppen erstellen— Wenn Sie keine benutzerdefinierte Cluster-Parametergruppe haben, die Sie verwenden können, erstellen Sie eine.

2. Ändern eines Amazon DocumentDB-Clusters— Ändern Sie Ihren Cluster so, dass er die benutzerdefinierte Cluster-Parametergruppe verwendet.

2. Bestimmen Sie den aktuellen Wert des **`tls`**-Cluster-Parameters.

   Wenn Sie weitere Informationen zu dieser Cluster-Parametergruppe wünschen, verwenden Sie die Operation `describe-db-cluster-parameters` mit den folgenden Parametern.

   - **`--db-cluster-parameter-group-name`** – Erforderlich. Verwenden Sie den Namen der Cluster-Parametergruppe aus der Ausgabe des vorherigen Befehls.
   - **`--query`**— Fakultativ. Eine Abfrage, mit der die Ausgabe auf die gewünschten Felder beschränkt wird – in diesem Fall `ParameterName`, `ParameterValue`, `AllowedValues` und `ApplyType`.

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --query 'Parameters[*].
[ParameterName,ParameterValue,AllowedValues,ApplyType]'
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "audit_logs",
        "disabled",
        "enabled,disabled",
        "dynamic"
    ],
    [
        "tls",
        "disabled",
        "disabled,enabled,fips-140-3",
        "static"
    ],
    [
        "ttl_monitor",
        "enabled",
        "disabled,enabled",
        "dynamic"
    ]
]
```

3. Ändern Sie den Wert des **`tls`**-Cluster-Parameters.

   Wenn der Wert für `tls` nicht der benötigte Wert ist, ändern Sie ihn für diese Cluster-Parametergruppe. Um den Wert des `tls`-Cluster-Parameters zu ändern, verwenden Sie die Operation`modify-db-cluster-parameter-group` mit den folgenden Parametern.

   - **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der zu ändernden Cluster-Parametergruppe. Dabei darf es sich nicht um eine `default.*`-Cluster-Parametergruppe handeln.
   - **`--parameters`** – Erforderlich. Eine Liste der zu ändernden Parameter der Cluster-Parametergruppe.
     - **`ParameterName`** – Erforderlich. Der Name des zu ändernden Cluster-Parameters.
     - **`ParameterValue`** – Erforderlich. Der neue Wert für diesen Cluster-Parameter. Muss einer der `AllowedValues` des Cluster-Parameters sein.
       - **`enabled`**— Der Cluster akzeptiert nur sichere Verbindungen mit TLS.
       - **`disabled`**— Der Cluster akzeptiert keine sicheren Verbindungen mit TLS.
       - **`fips-140-3`**— Der Cluster akzeptiert nur sichere Verbindungen gemäß den Anforderungen der Veröffentlichung 140-3 der Federal Information Processing Standards (FIPS). Dies wird erst ab Amazon DocumentDB 5.0-Clustern (Engine-Version 3.0.3727) in diesen Regionen unterstützt: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west
     - **`ApplyMethod`**— Wann diese Änderung angewendet werden soll. Für statische Cluster-Parameter wie `tle` muss dieser Wert `pending-reboot` lauten.
       - **`pending-reboot`**— Die Änderung wird erst auf eine Instanz angewendet, nachdem sie neu gestartet wurde. Sie müssen jede Cluster-Instance einzeln neu starten, damit die Änderung für alle Instances des Clusters übernommen wird.

   Der folgende Code deaktiviert `tls` und übernimmt die Änderung für jede DB-Instance, wenn diese neu gestartet wird.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

Der folgende Code aktiviert `tls` und übernimmt die Änderung für jede DB-Instance, wenn diese neu gestartet wird.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters "ParameterName=tls,ParameterValue=enabled,ApplyMethod=pending-
reboot"
```

Der folgende Code aktiviert TLS mit und wendet die Änderung auf jede DB-Instance an`fips-140-3`, wenn sie neu gestartet wird.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom5-0-param-grp \
    --parameters
 "ParameterName=tls,ParameterValue=fips-140-3,ApplyMethod=pending-reboot"
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

4. Starten Sie Ihre Amazon DocumentDB DocumentDB-Instance neu.

   Starten Sie jede Instance des Clusters neu, sodass die Änderung für alle Instances im Cluster übernommen wird. Um eine Amazon DocumentDB DocumentDB-Instance neu zu starten, verwenden Sie den `reboot-db-instance` Vorgang mit dem folgenden Parameter:

   - **`--db-instance-identifier`** – Erforderlich. Die Kennung der neu zu startenden Instance.

   Der folgende Code startet die Instance `sample-db-instance` neu.

   Example

   Für Linux, macOS oder Unix:

   ```
   aws docdb reboot-db-instance \
         --db-instance-identifier sample-db-instance
   ```

Für Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-db-instance
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBInstance": {
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
        "PendingModifiedValues": {},
        "DBInstanceStatus": "rebooting",
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetIdentifier": "subnet-b3806e8f"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
```

```
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetIdentifier": "subnet-29ab1025"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupName": "default"
        },
        "PromotionTier": 2,
        "DBInstanceClass": "db.r5.4xlarge",
        "InstanceCreateTime": "2018-11-05T23:10:49.905Z",
        "PreferredBackupWindow": "00:00-00:30",
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-a50b-44d4-
b6a0-a177d5ff730b",
        "StorageEncrypted": true,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "EngineVersion": "3.6.0",
        "DbiResourceId": "db-SAMPLERESOURCEID",
        "DBInstanceIdentifier": "sample-cluster-instance-00",
        "Engine": "docdb",
        "AvailabilityZone": "us-east-1a",
        "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-cluster-
instance-00",
        "BackupRetentionPeriod": 1,
        "Endpoint": {
```

```
            "Address": "sample-cluster-instance-00.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        "DBClusterIdentifier": "sample-cluster"
    }
}
```

Es dauert einige Minuten, bis Ihre Instance neu gestartet wird. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Schlüsselverwaltung

Amazon DocumentDB verwendet AWS Key Management Service (AWS KMS), um Verschlüsselungsschlüssel abzurufen und zu verwalten. AWS KMS kombiniert sichere, hochverfügbare Hard- und Software, um ein für die Cloud skaliertes Schlüsselverwaltungssystem bereitzustellen. Mit AWS KMS können Sie Verschlüsselungsschlüssel erstellen und Richtlinien definieren, die steuern, wie diese Schlüssel verwendet werden können. AWS KMS unterstützt AWS CloudTrail, sodass Sie die Schlüsselverwendung überprüfen und sicherstellen können, dass die Schlüssel korrekt verwendet werden.

Ihre AWS KMS Schlüssel können in Kombination mit Amazon DocumentDB und unterstützten AWS Services wie Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), Amazon Elastic Block Store (Amazon EBS) und Amazon Redshift verwendet werden. Eine Liste der Services, die unterstützenAWS KMS, finden Sie unter Wie -AWSServices verwenden AWS KMS im AWS Key Management Service -Entwicklerhandbuch. Weitere Informationen zu AWS KMS finden Sie unter Was ist AWS Key Management Service?.

Identity and Access Management für Amazon DocumentDB

AWS Identity and Access Management (IAM) ist ein AWS-Service , mit dem ein Administrator den Zugriff auf - AWS Ressourcen sicher steuern kann. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon DocumentDB-Ressourcen zu nutzen. IAM ist ein AWS-Service , den Sie ohne zusätzliche Kosten verwenden können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in Amazon DocumentDB .

Service-Benutzer – Wenn Sie den Amazon DocumentDB-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere Amazon DocumentDB-Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie nicht auf eine Funktion in Amazon DocumentDB zugreifen können, finden Sie weitere Informationen unter Fehlerbehebung für Amazon DocumentDB-Identität und -Zugriff.

Service-Administrator – Wenn Sie in Ihrem Unternehmen für Amazon DocumentDB-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf Amazon DocumentDB . Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon DocumentDB-Funktionen und -Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Amazon DocumentDB verwenden kann, finden Sie unter Funktionsweise von Amazon DocumentDB mit IAM.

IAM-Administrator – Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon DocumentDB verfassen können. Beispiele für identitätsbasierte Amazon DocumentDB-Richtlinien, die Sie in IAM verwenden können, finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB.

Authentifizierung mit Identitäten

Die Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten bei anmelden. Sie müssen als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle authentifiziert (bei angemeldet AWS) sein.

Sie können sich bei AWS als Verbundidentität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt werden. AWS IAM Identity Center (IAM Identity Center)-Benutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie AWS über einen Verbund auf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, um welchen Benutzertyp es sich handelt, können Sie sich bei der AWS Management Console oder im - AWS Zugriffsportal anmelden. Weitere Informationen zur Anmeldung bei AWSfinden Sie unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung - Benutzerhandbuch.

Wenn Sie AWS programmgesteuert auf zugreifen, AWS stellt ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (Command Line Interface, CLI) bereit, um Ihre Anforderungen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine - AWS Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenständigen Signieren von Anforderungen finden Sie unter Signieren von AWS API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. empfiehlt beispielsweise, AWS die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center - Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) in AWS im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet und Sie melden sich mit der E-Mail-Adresse und dem Passwort an, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Fordern Sie als bewährte Methode menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, auf, den Verbund mit einem Identitätsanbieter zu verwenden, um AWS-Services mithilfe temporärer Anmeldeinformationen auf zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, ein Web-Identitätsanbieter, die AWS Directory Service, das Identity-Center-Verzeichnis oder jeder Benutzer, der mit AWS-Services Anmeldeinformationen auf zugreift, die über eine Identitätsquelle bereitgestellt werden. Wenn Verbundidentitäten auf zugreifen AWS-Konten, übernehmen sie Rollen und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen oder eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie für alle Ihre AWS-Konten und Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Erstellen eines IAM-Benutzers (anstatt einer Rolle) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI - oder AWS -API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Verwenden von IAM-Rollen im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können AWS-ServicesSie jedoch eine Richtlinie direkt an

eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

- Serviceübergreifender Zugriff – Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen in auszuführen AWS, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung AWS-Service , Anforderungen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Service eine Anfrage erhält, die Interaktionen mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle: Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Serviceverknüpfte Rolle – Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Anwendungen, die auf Amazon EC2 ausgeführt werden – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und - AWS CLI oder AWS -API-Anforderungen stellen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine - AWS Rolle zuzuweisen und sie für alle ihre Anwendungen verfügbar zu machen, erstellen Sie ein Instance-Profil, das der Instance zugeordnet ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden einer IAM-

Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter Erstellen einer IAM-Rolle (anstatt eines Benutzers) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff in , AWS indem Sie Richtlinien erstellen und sie an AWS Identitäten oder Ressourcen anfügen. Eine Richtlinie ist ein Objekt in , AWS das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus der AWS Management Console, der AWS CLIoder der AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem anfügen können AWS-Konto. Verwaltete Richtlinien umfassen AWS -verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter Auswahl zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services AWS WAF, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter Zugriffssteuerungsliste (ACL) – Übersicht (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs)– SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in angeben AWS Organizations. AWS Organizations ist ein Service zum Gruppieren und zentralen Verwalten mehrerer AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Die SCP beschränkt Berechtigungen für Entitäten in Mitgliedskonten, einschließlich jeder Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter Funktionsweise von SCPs im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Wie AWS bestimmt, ob eine Anforderung zugelassen werden soll, wenn mehrere Richtlinientypen beteiligt sind, erfahren Sie unter Logik zur Richtlinienbewertung im IAM-Benutzerhandbuch.

Funktionsweise von Amazon DocumentDB mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon DocumentDB zu verwalten, erfahren Sie, welche IAM-Funktionen Sie mit Amazon DocumentDB verwenden können.

IAM-Funktionen, die Sie mit Amazon DocumentDB verwenden können

IAM-Feature	Instance-basierte Cluster	Elastic Cluster
Identitätsbasierte Richtlinien	Ja	Ja
Ressourcenbasierte Richtlinien	Nein	Nein
Richtlinienaktionen	Ja	Ja
Richtlinienressourcen	Ja	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja	Ja
ACLs	Nein	Nein
ABAC (Tags in Richtlinien)	Teilweise	Ja
Temporäre Anmeldeinformationen	Ja	Ja
Hauptberechtigungen	Ja	Ja
Servicerollen	Ja	Ja
Service-verknüpfte Rollen	Nein	Ja

Einen Überblick über das Zusammenwirken von Amazon DocumentDB und anderen - AWS Services mit den meisten IAM-Funktionen finden Sie unter -AWS Services, die mit IAM funktionieren im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien für Amazon DocumentDB

Unterstützt Richtlinien auf Identitätsbasis.	Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern,

welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Beispiele für identitätsbasierte Amazon DocumentDB-Richtlinien finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB.

Ressourcenbasierte Richtlinien in Amazon DocumentDB

Unterstützt ressourcenbasierte Richtlinien	Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource in unterschiedlichen befinden AWS-Konten, muss ein IAM-Administrator im vertrauenswürdigen Konto auch der Prinzipal-Entität (Benutzer oder Rolle) die Berechtigung für den Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal

in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden im IAM-Benutzerhandbuch.

Richtlinienaktionen für Amazon DocumentDB

Unterstützt Richtlinienaktionen	Ja

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Eine Liste der RDS-Aktionen finden Sie unter Von Amazon Relational Database Service definierte Aktionen in der Service-Autorisierungs-Referenz.
> Informationen zum Anzeigen von Richtlinienaktionen für Amazon DocumentDB-Eleastic-Cluster finden Sie unter Von elastischen Amazon DocumentDB-Clustern definierte Aktionen in der Service-Autorisierungs-Referenz.

Richtlinienaktionen in Amazon DocumentDB verwenden das folgende Präfix vor der Aktion:

```
aws
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "aws:action1",
      "aws:action2"
         ]
```

Beispiele für identitätsbasierte Amazon DocumentDB-Richtlinien finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB.

Richtlinienressourcen für Amazon DocumentDB

Unterstützt Richtlinienressourcen	Ja

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Eine Liste der RDS-Ressourcentypen und ihrer ARNs finden Sie unter Von Amazon Relational Database Service definierte Ressourcen in der Service-Autorisierungs-Referenz.

Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von Amazon Relational Database Service definierte Aktionen. Informationen zum Anzeigen von Ressourcentypen für Amazon DocumentDB-Eleastic-Cluster finden Sie unter Von elastischen Amazon DocumentDB-Clustern definierte Ressourcentypen in der Service-Autorisierungs-Referenz.

Beispiele für identitätsbasierte Amazon DocumentDB-Richtlinien finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB.

Richtlinienbedingungsschlüssel für Amazon DocumentDB

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Informationen zum Anzeigen aller AWS globalen Bedingungsschlüssel finden Sie unter AWS Globale Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Eine Liste der RDS-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Relational Database Service in der Service-Autorisierungs-Referenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon Relational Database Service definierte Aktionen.
> Informationen zum Anzeigen von Bedingungsschlüsseln für Amazon DocumentDB-Eleastic-Cluster finden Sie unter Bedingungsschlüssel für elastische Amazon DocumentDB-Cluster in der Service-Autorisierungs-Referenz.

Beispiele für identitätsbasierte Amazon DocumentDB-Richtlinien finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB.

ACLs in Amazon DocumentDB

Unterstützt ACLs	Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon DocumentDB

> **Note**
>
> ABAC wird nur teilweise für Instance-basierte Cluster unterstützt, aber für elastische Cluster.

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWSdiese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anfügen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Was ist ABAC? im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit Amazon DocumentDB

Unterstützt temporäre Anmeldeinformationen	Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich der , die mit temporären Anmeldeinformationen AWS-Services funktionieren, finden Sie unter AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich AWS Management Console mit einer anderen Methode als einem Benutzernamen und einem Passwort bei der anmelden. Wenn Sie beispielsweise AWS über den SSO-Link (Single Sign-On) Ihres Unternehmens auf zugreifen, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln zu einer Rolle (Konsole) im IAM-Benutzerhandbuch.

Sie können temporäre Anmeldeinformationen manuell mit der AWS CLI oder der AWS API erstellen. Sie können diese temporären Anmeldeinformationen dann verwenden, um auf zuzugreifen AWS. AWS empfohlen, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipal-Berechtigungen für Amazon DocumentDB

Unterstützt Forward Access Sessions (FAS)	Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen in auszuführen AWS, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung AWS-Service , Anfragen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Service eine Anfrage erhält, die Interaktionen mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für Amazon DocumentDB

Unterstützt Servicerollen	Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle könnte die Funktionalität von Amazon DocumentDB beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon DocumentDB dazu Anleitungen gibt.

Serviceverknüpfte Rollen für Amazon DocumentDB

> **Note**
>
> Serviceverknüpfte Rollen werden für Instance-basierte Cluster nicht unterstützt, für elastische Cluster jedoch.

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon DocumentDB-Ressourcen. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von Amazon DocumentDB definiert werden, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Relational Database Service in der Service-Autorisierungs-Referenz.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Amazon DocumentDB-Konsole

- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon DocumentDB-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit von AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu erteilen, verwenden Sie die von AWS verwalteten Richtlinien, die Berechtigungen für viele häufige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die für Ihre Anwendungsfälle spezifisch sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten:Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn sie über eine bestimmte verwendet werden AWS-Service, z. B. AWS CloudFormation. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere

und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.

- Multi-Faktor-Authentifizierung (MFA) erforderlich – Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon DocumentDB-Konsole

Um auf die Amazon DocumentDB-Konsole (mit MongoDB-Kompatibilität) zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon DocumentDB-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Für Benutzer, die nur Aufrufe an die AWS CLI oder die AWS API durchführen, müssen Sie keine Mindestberechtigungen für die Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen weiterhin die Amazon DocumentDB-Konsole verwenden können, fügen Sie den Entitäten auch die von Amazon DocumentDB *ConsoleAccess* oder *ReadOnly* AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI oder AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Fehlerbehebung für Amazon DocumentDB-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon DocumentDB und IAM auftreten können.

Themen

- Ich bin nicht autorisiert, eine Aktion in Amazon DocumentDB auszuführen
- Ich bin nicht autorisiert, iam durchzuführen:PassRole
- Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Amazon DocumentDB-Ressourcen gewähren

Ich bin nicht autorisiert, eine Aktion in Amazon DocumentDB auszuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `aws:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aws:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `aws:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht autorisiert, iam durchzuführen:PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Ausführen der `iam:PassRole` Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon DocumentDB übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine vorhandene Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder serviceverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon DocumentDB auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Amazon DocumentDB-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffssteuerungslisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon DocumentDB diese Funktionen unterstützt, finden Sie unter Funktionsweise von Amazon DocumentDB mit IAM.
- Informationen zum Gewähren des Zugriffs auf Ihre Ressourcen in Ihrem Besitz finden AWS-Konten Sie unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , das Sie besitzen im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre -Ressourcen gewähren AWS-Konten, finden Sie unter Gewähren von Zugriff auf im AWS-Konten Besitz von Dritten im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien im IAM-Benutzerhandbuch.

Verwalten von Zugriffsberechtigungen für Ihre Amazon DocumentDB-Ressourcen

Jede AWS Ressource gehört einem AWS-Konto, und Berechtigungen zum Erstellen oder Zugreifen auf die Ressourcen werden durch Berechtigungsrichtlinien geregelt. Ein Kontoadministrator kann IAM-Identitäten (d. h. Benutzern, Gruppen und Rollen) Berechtigungsrichtlinien zuweisen. Einige -Services (z. B. AWS Lambda) unterstützen auch das Anfügen von Berechtigungsrichtlinien an Ressourcen.

> **Note**
>
> Ein Kontoadministrator (oder Administratorbenutzer) ist ein Benutzer mit Administratorberechtigungen. Weitere Informationen finden Sie unter Bewährte Methoden für IAM im IAM-Benutzerhandbuch.

Themen

- Amazon DocumentDB-Ressourcen und -Vorgänge
- Grundlegendes zum Eigentum an Ressourcen
- Verwalten des Zugriffs auf Ressourcen
- Angeben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale
- Angeben von Bedingungen in einer Richtlinie

Amazon DocumentDB-Ressourcen und -Vorgänge

In Amazon DocumentDB ist die primäre Ressource ein Cluster . Amazon DocumentDB unterstützt andere Ressourcen, die mit der primären Ressource verwendet werden können, z. B. Instances , Parametergruppen und Ereignisabonnements . Diese Ressourcen werden als Unterressourcen bezeichnet.

Diese Ressourcen und Unterressourcen sind eindeutigen Amazon-Ressourcennamen (ARNs) zugeordnet (siehe Tabelle unten).

Ressourcentyp	ARN-Format
Cluster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster:` *`db-cluster-name`*
Cluster-Parameterg ruppe	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-pg:` *`cluster-parameter-group-name`*
Cluster-Snapshot	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-snapshot:` *`cluster-snapshot-name`*
Instance	`arn:aws:rds:` *`region`*`:`*`account-id`* `:db:`*`db-instance-name`*

Ressourcentyp	ARN-Format
Sicherheitsgruppe	`arn:aws:rds:` *region*`:`*account-id* `:secgrp:`*security-group-name*
Subnetzgruppe	`arn:aws:rds:` *region*`:`*account-id* `:subgrp:`*subnet-group-name*

Amazon DocumentDB bietet eine Reihe von Operationen für die Arbeit mit den Amazon DocumentDB-Ressourcen. Eine Liste der verfügbaren Operationen finden Sie unter Aktionen.

Grundlegendes zum Eigentum an Ressourcen

Ein Ressourcenbesitzer ist das AWS-Konto , das eine Ressource erstellt hat. Das heißt, der Ressourceneigentümer ist der AWS-Konto der Prinzipal-Entität (das Stammkonto, ein IAM-Benutzer oder eine IAM-Rolle), die die Anforderung authentifiziert, die die Ressource erstellt. Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Wenn Sie die Root-Konto-Anmeldeinformationen Ihres verwenden, AWS-Konto um eine Amazon DocumentDB-Ressource zu erstellen, z. B. eine -Instance, AWS-Konto ist Ihr der Eigentümer der Amazon DocumentDB-Ressource.
- Wenn Sie einen IAM-Benutzer in Ihrem erstellen AWS-Konto und diesem Berechtigungen zum Erstellen von Amazon DocumentDB-Ressourcen erteilen, kann der Benutzer Amazon DocumentDB-Ressourcen erstellen. Eigentümer der Amazon DocumentDB-Ressourcen ist jedoch Ihr AWS-Konto, zu dem der Benutzer gehört.
- Wenn Sie in Ihrem eine IAM-Rolle AWS-Konto mit Berechtigungen zum Erstellen von Amazon DocumentDB-Ressourcen erstellen, kann jeder, der die Rolle übernimmt, Amazon DocumentDB-Ressourcen erstellen. Eigentümer der Amazon DocumentDB-Ressourcen ist das , AWS-Kontozu dem die Rolle gehört.

Verwalten des Zugriffs auf Ressourcen

Eine Berechtigungsrichtlinie beschreibt, wer Zugriff auf welche Objekte hat. Im folgenden Abschnitt werden die verfügbaren Optionen zum Erstellen von Berechtigungsrichtlinien erläutert.

> **Note**
>
> In diesem Abschnitt wird die Verwendung von IAM im Kontext von Amazon DocumentDB beschrieben. Er enthält keine detaillierten Informationen über den IAM-Service. Eine umfassende IAM-Dokumentation finden Sie unter Was ist IAM? im IAM-Benutzerhandbuch. Informationen zur IAM-Richtliniensyntax und Beschreibungen finden Sie in der AWSIAM Richtlinienreferenz im IAM-Benutzerhandbuch.

An eine IAM-Identität angefügte Richtlinien werden als identitätsbasierte Richtlinien (IAM-Richtlinien) bezeichnet. An Ressourcen angehängte Richtlinien werden als ressourcenbasierte Richtlinien bezeichnet. Amazon DocumentDB unterstützt nur identitätsbasierte Richtlinien (IAM-Richtlinien).

Themen

- Identitätsbasierte Richtlinien (IAM-Richtlinien)
- Ressourcenbasierte Richtlinien

Identitätsbasierte Richtlinien (IAM-Richtlinien)

Richtlinien können IAM-Identitäten angefügt werden. Sie können z. B. Folgendes tun:

- Einem Benutzer oder einer Gruppe in Ihrem Konto eine Berechtigungsrichtlinie zuweisen – Ein Kontoadministrator kann eine Berechtigungsrichtlinie verwenden, die einem bestimmten Benutzer zugeordnet ist, um diesem Benutzer Berechtigungen zum Erstellen einer Amazon DocumentDB-Ressource zu erteilen, z. B. einer Instance.
- Einer Rolle eine Berechtigungsrichtlinie zuweisen (kontoübergreifende Berechtigungen gewähren) – Sie können einer IAM-Rolle eine identitätsbasierte Berechtigungsrichtlinie zuweisen, um kontoübergreifende Berechtigungen zu erteilen. Ein Administrator kann beispielsweise eine Rolle erstellen, um einem anderen AWS-Konto oder einem - AWS Service kontoübergreifende Berechtigungen zu erteilen:
  1. Der Administrator von Konto A erstellt eine IAM-Rolle und fügt ihr eine Berechtigungsrichtlinie an, die Berechtigungen für Ressourcen in Konto A erteilt.
  2. Der Administrator von Konto A weist der Rolle eine Vertrauensrichtlinie zu, die Konto B als den Prinzipal identifiziert, der die Rolle übernehmen kann.
  3. Der Administrator von Konto B kann dann Berechtigungen zur Übernahme der Rolle an alle Benutzer in Konto B delegieren. Dadurch können die Benutzer in Konto B Ressourcen in Konto

A erstellen oder darauf zugreifen. Der Prinzipal in der Vertrauensrichtlinie kann auch ein - AWS Service-Prinzipal sein, wenn Sie einem - AWS Service Berechtigungen zur Übernahme der Rolle erteilen möchten.

Weitere Informationen zum Delegieren von Berechtigungen mithilfe von IAM finden Sie unter Zugriffsverwaltung im IAM-Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die es dem Benutzer mit der -ID ermöglicht`123456789012`, Instances für Ihr zu erstellen AWS-Konto. Die neue Instance muss eine Optionsgruppe und eine Parametergruppe verwenden, die mit `default` beginnt, und sie muss die Subnetzgruppe `default` verwenden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:pg:cluster-pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ]
        }
    ]
}
```

Weitere Informationen zur Verwendung identitätsbasierter Richtlinien mit Amazon DocumentDB finden Sie unter Verwenden von identitätsbasierten Richtlinien (IAM-Richtlinien) für Amazon DocumentDB. Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie im Thema Identitäten (Benutzer, Gruppen und Rollen) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Andere -Services wie Amazon Simple Storage Service (Amazon S3) unterstützen auch ressourcenbasierte Berechtigungsrichtlinien. Beispielsweise können Sie einem Amazon S3-Bucket

eine Richtlinie zuweisen, um die Zugriffsberechtigungen für diesen Bucket zu verwalten. Amazon DocumentDB unterstützt keine ressourcenbasierten Richtlinien.

Angeben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale

Für jede Amazon DocumentDB-Ressource (siehe Amazon DocumentDB-Ressourcen und -Vorgänge) definiert der Service eine Reihe von API-Operationen. Weitere Informationen finden Sie unter Aktionen. Um Berechtigungen für diese API-Operationen zu erteilen, definiert Amazon DocumentDB eine Reihe von Aktionen, die Sie in einer Richtlinie angeben können. Für das Durchführen einer API-Operation können Berechtigungen für mehrere Aktionen erforderlich sein.

Grundlegende Richtlinienelemente:

- Ressource – In einer Richtlinie wird der Amazon-Ressourcenname (ARN) zur Identifizierung der Ressource verwendet, für die die Richtlinie gilt.
- Aktion – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenoperationen an, die Sie zulassen oder verweigern möchten. Die `rds:DescribeDBInstances`-Berechtigung erteilt dem Benutzer zum Beispiel Berechtigungen zum Ausführen der `DescribeDBInstances`-Operation.
- Auswirkung – Die von Ihnen festgelegte Auswirkung, wenn der Benutzer die jeweilige Aktion anfordert – entweder „allow“ (Zugriffserlaubnis) oder „deny“ (Zugriffsverweigerung). Wenn Sie den Zugriff auf eine Ressource nicht ausdrücklich gestatten ("Allow"), wird er automatisch verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So können Sie sicherstellen, dass Benutzer nicht darauf zugreifen können, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.
- Prinzipal – In identitätsbasierten Richtlinien (IAM-Richtlinien) ist der Benutzer, dem die Richtlinie zugewiesen ist, automatisch der Prinzipal. In ressourcenbasierten Richtlinien müssen Sie den Benutzer, das Konto, den Service oder die sonstige Entität angeben, die die Berechtigungen erhalten soll (gilt nur für ressourcenbasierte Richtlinien). Amazon DocumentDB unterstützt keine ressourcenbasierten Richtlinien.

Weitere Informationen zur Syntax und zu Beschreibungen von IAM-Richtlinien finden Sie in der AWS -IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Eine Tabelle mit allen Amazon DocumentDB-API-Aktionen und den Ressourcen, für die sie gelten, finden Sie unter Amazon DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen.

Angeben von Bedingungen in einer Richtlinie

Beim Erteilen von Berechtigungen können Sie mithilfe der IAM-Richtliniensyntax die Bedingungen angeben, unter denen die Richtlinie wirksam werden soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum gilt. Weitere Informationen zum Angeben von Bedingungen in einer Richtliniensyntax finden Sie im Thema Bedingung im IAM Benutzerhandbuch.

Bedingungen werden mithilfe vordefinierter Bedingungsschlüssel formuliert. Amazon DocumentDB besitzt keine servicespezifischen Kontextschlüssel, die in einer IAM-Richtlinie verwendet werden können. Eine Liste der globalen Kontextschlüssel für Bedingungen, die für alle Services verfügbar sind, finden Sie unter Verfügbare Schlüssel für Bedingungen im IAM-Benutzerhandbuch.

Verwenden von identitätsbasierten Richtlinien (IAM-Richtlinien) für Amazon DocumentDB

> **Important**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS geteilt wird. Amazon DocumentDB- AWS CLIKonsolen- und -API-Aufrufe werden als Aufrufe der Amazon-RDS-API protokolliert.
> Wir empfehlen Ihnen, zunächst die einführenden Themen zu lesen, in denen die grundlegenden Konzepte und Optionen erläutert werden, mit denen Sie den Zugriff auf Ihre Amazon DocumentDB-Ressourcen verwalten können. Weitere Informationen finden Sie unter Verwalten von Zugriffsberechtigungen für Ihre Amazon DocumentDB-Ressourcen.

In diesem Thema finden Sie Beispiele für identitätsbasierte Richtlinien, in denen ein Kontoadministrator den IAM-Identitäten (Benutzer, Gruppen und Rollen) Berechtigungsrichtlinien anfügen kann.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
```

```
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:pg:cluster-pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ]
        }
    ]
}
```

Die Richtlinie ist ein einzelnes Statement, das die folgenden Berechtigungen für den IAM-Benutzer bestimmt:

- Die Richtlinie ermöglicht es dem IAM-Benutzer, eine Instance mit der Aktion CreateDBInstance zu erstellen (dies gilt auch für die -create-db-instance AWS CLI Operation und die AWS Management Console).
- Das Element `Resource` gibt an, dass der Benutzer auf oder mit Ressourcen Aktionen ausführen kann. Sie geben Ressourcen über einen Amazon-Ressourcennamen (ARN) an. Dieser ARN enthält den Namen des Services, zu dem die Ressource gehört (`rds`), die AWS-Region (`*` zeigt in diesem Beispiel eine beliebige Region an), die Benutzerkontonummer (`123456789012` ist die Benutzer-ID in diesem Beispiel) und den Typ der Ressource.

  Das `Resource`-Element im Beispiel gibt für den Benutzer die folgenden richtlinienbezogenen Einschränkungen für die Ressourcen an:

  - Die Instance-Kennung für die neue Instance muss mit `test` beginnen (zum Beispiel `testCustomerData1`, `test-region2-data`).
  - Die Parametergruppe für die neue Instance muss mit `default` beginnen.
  - Die Subnetzgruppe für die neue Instance muss mit `default` beginnen.

Das Element `Principal` ist in der Richtlinie nicht angegeben, da in identitätsbasierten Richtlinien die Angabe des Prinzipals als Empfänger der Berechtigung nicht erforderlich ist. Wenn Sie einem Benutzer eine Richtlinie zuweisen, ist der Benutzer automatisch der Prinzipal. Wird die Berechtigungsrichtlinie einer IAM-Rolle angefügt, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen Amazon DocumentDB-API-Operationen und den Ressourcen, für die sie gelten, finden Sie unter Amazon DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen.

Erforderliche Berechtigungen für die Verwendung der Amazon DocumentDB-Konsole

Damit ein Benutzer mit der Amazon DocumentDB-Konsole arbeiten kann, muss dieser Benutzer über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen ermöglichen es dem Benutzer, die Amazon DocumentDB-Ressourcen für sein zu beschreiben AWS-Konto und andere zugehörige Informationen bereitzustellen, einschließlich Amazon EC2-Sicherheits- und Netzwerkinformationen.

Wenn Sie eine IAM-Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Benutzer mit dieser IAM-Richtlinie. Um sicherzustellen, dass diese Benutzer weiterhin die Amazon DocumentDB-Konsole verwenden können, fügen Sie dem Benutzer auch die `AmazonDocDBConsoleFullAccess` verwaltete Richtlinie an, wie unter beschriebenAWS Von verwaltete Richtlinien für Amazon DocumentDB.

Für Benutzer, die nur Aufrufe an die AWS CLI oder die Amazon DocumentDB-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen.

Beispiele für vom Kunden verwaltete Richtlinien

In diesem Abschnitt finden Sie Beispiele für Benutzerrichtlinien, die Berechtigungen für verschiedene Amazon DocumentDB-Aktionen gewähren. Diese Richtlinien funktionieren, wenn Sie Amazon DocumentDB-API-Aktionen, AWS SDKs oder die verwenden AWS CLI. Bei Verwendung der Konsole müssen Sie zusätzliche konsolenspezifische Berechtigungen erteilen, die im Abschnitt Erforderliche Berechtigungen für die Verwendung der Amazon DocumentDB-Konsole erläutert werden.

Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) und Amazon Neptune gemeinsam genutzt wird.

> **Note**
>
> Alle Beispiele verwenden die Region USA Ost (Nord-Virginia) (`us-east-1`) und enthalten fiktive Konto-IDs .

Beispiele

Beispiel 1: Einem Benutzer erlauben, jede Beschreibungsaktion für jede Amazon DocumentDB-Ressource auszuführen

Die folgende Berechtigungsrichtlinie gewährt Berechtigungen für einen Benutzer, alle Aktionen auszuführen, die mit beginne `Describe`. Diese Aktionen zeigen Informationen zu einer Amazon DocumentDB-Ressource, z. B. einer -Instance. Das Platzhalterzeichen (*) im -`Resource`Element gibt an, dass die Aktionen für alle Amazon DocumentDB-Ressourcen zulässig sind, die dem Konto gehören.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowRDSDescribe",
         "Effect":"Allow",
         "Action":"rds:Describe*",
         "Resource":"*"
      }
   ]
}
```

Beispiel 2: Einen Nutzer davon abhalten, eine Instance zu löschen

Die folgenden Berechtigungsrichtlinien erteilen Berechtigungen, um einen Benutzer davon abzuhalten, eine bestimmte Instance zu löschen. Beispielsweise möchten Sie jedem Benutzer, der kein Administrator ist, verbieten, Ihre Produktions-Instances zu löschen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyDelete1",
         "Effect":"Deny",
```

```
         "Action":"rds:DeleteDBInstance",
         "Resource":"arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
      }
   ]
}
```

Beispiel 3: Verhindern, dass ein Benutzer einen Cluster erstellt, es sei denn, die Speicherverschlüsselung ist aktiviert

Die folgende Berechtigungsrichtlinie verweigert einem Benutzer das Erstellen eines Amazon DocumentDB-Clusters, es sei denn, die Speicherverschlüsselung ist aktiviert.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PreventUnencryptedDocumentDB",
         "Effect": "Deny",
         "Action": "RDS:CreateDBCluster",
         "Condition": {
         "Bool": {
         "rds:StorageEncrypted": "false"
      },
         "StringEquals": {
         "rds:DatabaseEngine": "docdb"
         }
      },
      "Resource": "*"
      }
   ]
}
```

AWS Von verwaltete Richtlinien für Amazon DocumentDB

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als selbst Richtlinien zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere von AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken häufige Anwendungsfälle ab und sind in Ihrem AWS Konto verfügbar. Weitere Informationen zu von AWS verwalteten Richtlinien finden Sie unter Von AWS verwaltete Richtlinien im AWS Identity and Access Management-Benutzerhandbuch.

AWS -Services verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Services fügen einer von AWS verwalteten Richtlinie gelegentlich zusätzliche Berechtigungen hinzu, um neue Funktionen zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am wahrscheinlichsten, wenn eine neue Funktion gestartet wird oder wenn neue Vorgänge verfügbar werden. Services entfernen keine Berechtigungen aus einer von AWS verwalteten Richtlinie, sodass Richtlinienaktualisierungen Ihre vorhandenen Berechtigungen nicht beeinträchtigen.

Darüber hinaus AWS unterstützt verwaltete Richtlinien für Auftragsfunktionen, die sich über mehrere Services erstrecken. Die von `ViewOnlyAccess` AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf viele - AWS Services und -Ressourcen. Wenn ein Service ein neues Feature startet, AWS fügt schreibgeschützte Berechtigungen für neue Vorgänge und Ressourcen hinzu. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie unter Von AWS verwaltete Richtlinien für Auftragsfunktionen im AWS Benutzerhandbuch für Identity and Access Management.

Die folgenden AWS verwalteten Richtlinien, die Sie Benutzern in Ihrem Konto anfügen können, gelten speziell für Amazon DocumentDB :

- AmazonDocDBFullAccess – Gewährt vollen Zugriff auf alle Amazon DocumentDB-Ressourcen für das AWS Stammkonto.
- AmazonDocDBReadOnlyAccess – Gewährt schreibgeschützten Zugriff auf alle Amazon DocumentDB-Ressourcen für das AWS Stammkonto.
- AmazonDocDBConsoleFullAccess – Gewährt vollen Zugriff auf die Verwaltung von Amazon DocumentDB- und Amazon DocumentDB-Elastic-Cluster-Ressourcen mithilfe der AWS Management Console.
- AmazonDocDBElasticReadOnlyAccess – Gewährt schreibgeschützten Zugriff auf alle Amazon DocumentDB-Cluster-Ressourcen für das AWS Stammkonto.
- AmazonDocDBElasticFullAccess – Gewährt vollen Zugriff auf alle elastischen Amazon DocumentDB-Cluster-Ressourcen für das AWS Stammkonto.

AmazonDocDBFullAccess

Diese Richtlinie gewährt Administratorberechtigungen, die einem Prinzipal vollen Zugriff auf alle Amazon DocumentDB-Aktionen ermöglichen. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Die Amazon DocumentDB-Berechtigungen erlauben alle Amazon DocumentDB-Aktionen.
- Einige der Amazon-EC2-Berechtigungen in dieser Richtlinie sind erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch wird sichergestellt, dass Amazon DocumentDB die Ressourcen erfolgreich mit einem Cluster verwenden kann. Die restlichen Amazon EC2-Berechtigungen in dieser Richtlinie ermöglichen es Amazon DocumentDB, AWS Ressourcen zu erstellen, die erforderlich sind, damit Sie eine Verbindung zu Ihren Clustern herstellen können.
- Die Amazon DocumentDB-Berechtigungen werden während API-Aufrufen verwendet, um die übergebenen Ressourcen in einer Anforderung zu validieren. Sie sind erforderlich, damit Amazon DocumentDB den übergebenen Schlüssel mit dem Amazon DocumentDB-Cluster verwenden kann.
- Die CloudWatch Protokolle sind erforderlich, damit Amazon DocumentDB sicherstellen kann, dass die Protokollbereitstellungsziele erreichbar sind und dass sie für die Verwendung von Broker-Protokollen gültig sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:DeleteDBCluster",
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
                "rds:DeleteDBSubnetGroup",
```

```
                "rds:DeleteEventSubscription",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
```

```
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": "iam:CreateServiceLinkedRole",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "rds.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDBReadOnlyAccess

Diese Richtlinie gewährt schreibgeschützte Berechtigungen, die es Benutzern ermöglichen, Informationen in Amazon DocumentDB anzuzeigen. Prinzipale, denen diese Richtlinie angefügt ist, können keine Aktualisierungen vornehmen oder bestehende Ressourcen löschen und auch keine neuen Amazon DocumentDB-Ressourcen erstellen. Prinzipale mit diesen Berechtigungen können beispielsweise die Liste der Cluster und Konfigurationen, die mit ihrem Konto verknüpft sind, einsehen, aber nicht die Konfiguration oder Einstellungen von Clustern ändern. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Mit Amazon DocumentDB-Berechtigungen können Sie Amazon DocumentDB-Ressourcen auflisten, beschreiben und Informationen zu ihnen abrufen.
- Amazon EC2-Berechtigungen werden verwendet, um die Amazon VPC, Subnetze, Sicherheitsgruppen und ENIs zu beschreiben, die einem Cluster zugeordnet sind.
- Eine Amazon DocumentDB-Berechtigung wird verwendet, um den Schlüssel zu beschreiben, der dem Cluster zugeordnet ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
```

```
                "rds:DownloadDBLogFilePortion",
                "rds:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "kms:ListAliases",
                "kms:ListKeyPolicies"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "logs:DescribeLogStreams",
                "logs:GetLogEvents"
            ],
            "Effect": "Allow",
            "Resource": [
```

```
                "arn:aws:logs:*:*:log-group:/aws/rds/*:log-stream:*",
                "arn:aws:logs:*:*:log-group:/aws/docdb/*:log-stream:*"
            ]
        }
    ]
}
```

AmazonDocDBConsoleFullAccess

Gewährt vollen Zugriff auf die Verwaltung von Amazon DocumentDB-Ressourcen mithilfe der AWS Management Console für Folgendes:

- Die Amazon DocumentDB-Berechtigungen zum Zulassen aller Amazon DocumentDB- und Amazon DocumentDB-Cluster-Aktionen.
- Einige der Amazon-EC2-Berechtigungen in dieser Richtlinie sind erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch soll sichergestellt werden, dass Amazon DocumentDB die Ressourcen erfolgreich zur Bereitstellung und Wartung des Clusters verwenden kann. Die restlichen Amazon EC2-Berechtigungen in dieser Richtlinie ermöglichen es Amazon DocumentDB, AWS Ressourcen zu erstellen, die erforderlich sind, damit Sie eine Verbindung zu Ihren Clustern wie VPCEndpoint herstellen können.
- AWS KMS -Berechtigungen werden bei API-Aufrufen an verwendet, AWS KMS um die übergebenen Ressourcen in einer Anforderung zu validieren. Sie sind erforderlich, damit Amazon DocumentDB den übergebenen Schlüssel verwenden kann, um die Daten im Ruhezustand mit dem elastischen Amazon DocumentDB-Cluster zu verschlüsseln und zu entschlüsseln.
- Die CloudWatch Protokolle sind erforderlich, damit Amazon DocumentDB sicherstellen kann, dass die Ziele für die Protokollzustellung erreichbar sind und dass sie für die Verwendung von Prüfungs- und Profilerstellungsprotokollen gültig sind.
- Secrets-Manager-Berechtigungen sind erforderlich, um das angegebene Secret zu validieren und es als Administratorbenutzer für elastische Amazon DocumentDB-Cluster einzurichten.
- Amazon RDS-Berechtigungen sind für Amazon DocumentDB-Clusterverwaltungsaktionen erforderlich. Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS geteilt wird.
- SNS-Berechtigungen ermöglichen es Prinzipalen, Amazon Simple Notification Service (Amazon SNS)-Abonnements und -Themen zu abonnieren und Amazon SNS-Nachrichten zu veröffentlichen.
- IAM-Berechtigungen sind erforderlich, um die serviceverknüpften Rollen zu erstellen, die für die Veröffentlichung von Metriken und Protokollen erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbSids",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster",
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:CreateGlobalCluster",
                "rds:DeleteDBCluster",
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
```

```
                "rds:DeleteDBSubnetGroup",
                "rds:DeleteEventSubscription",
                "rds:DeleteGlobalCluster",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeGlobalClusters",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:ModifyGlobalCluster",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
                "rds:RemoveFromGlobalCluster",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
```

```
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DependencySids",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssignPrivateIpAddresses",
                "ec2:AssociateAddress",
                "ec2:AssociateRouteTable",
                "ec2:AssociateSubnetCidrBlock",
                "ec2:AssociateVpcCidrBlock",
                "ec2:AttachInternetGateway",
                "ec2:AttachNetworkInterface",
                "ec2:CreateCustomerGateway",
                "ec2:CreateDefaultSubnet",
                "ec2:CreateDefaultVpc",
                "ec2:CreateInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeInstances",
                "ec2:DescribeNatGateways",
```

```
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:ModifyVpcEndpoint",
                "kms:DescribeKey",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DocdbSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "rds.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DocdbElasticSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
```

```
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDBElasticReadOnlyAccess

Diese Richtlinie gewährt schreibgeschützte Berechtigungen, mit denen Benutzer Elastic-Cluster-Informationen in Amazon DocumentDB anzeigen können. Prinzipale, denen diese Richtlinie angefügt ist, können keine Aktualisierungen vornehmen oder bestehende Ressourcen löschen und auch keine neuen Amazon DocumentDB-Ressourcen erstellen. Prinzipale mit diesen Berechtigungen können beispielsweise die Liste der Cluster und Konfigurationen, die mit ihrem Konto verknüpft sind, einsehen, aber nicht die Konfiguration oder Einstellungen von Clustern ändern. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Mit den Berechtigungen für elastische Amazon DocumentDB-Cluster können Sie Ressourcen für elastische Amazon DocumentDB-Cluster auflisten, beschreiben und Informationen zu ihnen abrufen.
- CloudWatch -Berechtigungen werden verwendet, um Service-Metriken zu überprüfen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "docdb-elastic:ListClusters",
            "docdb-elastic:GetCluster",
            "docdb-elastic:ListClusterSnapshots",
            "docdb-elastic:GetClusterSnapshot",
            "docdb-elastic:ListTagsForResource"
         ],
         "Resource": "*"
      },
```

```
      {
         "Effect": "Allow",
         "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics",
            "cloudwatch:GetMetricStatistics"
         ],
         "Resource": "*"
      }
   ]
}
```

AmazonDocDBElasticFullAccess

Diese Richtlinie gewährt Administratorberechtigungen, die einem Prinzipal vollen Zugriff auf alle Amazon DocumentDB-Aktionen für den elastischen Amazon DocumentDB-Cluster ermöglichen.

Diese Richtlinie verwendet AWS Tags (https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html) innerhalb von Bedingungen, um den Zugriff auf -Ressourcen zu beschränken. Wenn Sie ein Secret verwenden, muss es mit einem Tag-Schlüssel `DocDBElasticFullAccess` und einem Tag-Wert gekennzeichnet sein. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, muss dieser mit einem Tag-Schlüssel `DocDBElasticFullAccess` und einem Tag-Wert gekennzeichnet sein.

Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Elastic-Cluster-Berechtigungen von Amazon DocumentDB erlauben alle Amazon DocumentDB-Aktionen.
- Einige der Amazon-EC2-Berechtigungen in dieser Richtlinie sind erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch soll sichergestellt werden, dass Amazon DocumentDB die Ressourcen erfolgreich zur Bereitstellung und Wartung des Clusters verwenden kann. Die restlichen Amazon EC2-Berechtigungen in dieser Richtlinie ermöglichen es Amazon DocumentDB, AWS Ressourcen zu erstellen, die erforderlich sind, damit Sie eine Verbindung zu Ihren Clustern herstellen können, z. B. zu einem VPC-Endpunkt.
- AWS KMS -Berechtigungen sind erforderlich, damit Amazon DocumentDB den übergebenen Schlüssel verwenden kann, um die Daten im Ruhezustand innerhalb des elastischen Amazon DocumentDB-Clusters zu verschlüsseln und zu entschlüsseln.

> **Note**
>
> Der vom Kunden verwaltete Schlüssel muss über ein Tag mit Schlüssel `DocDBElasticFullAccess` und einen Tag-Wert verfügen.

- SecretsManager -Berechtigungen sind erforderlich, um das angegebene Secret zu validieren und es als Administratorbenutzer für elastische Amazon DocumentDB-Cluster einzurichten.

> **Note**
>
> Das verwendete Secret muss ein Tag mit Schlüssel `DocDBElasticFullAccess` und einem Tag-Wert haben.

- IAM-Berechtigungen sind erforderlich, um die serviceverknüpften Rollen zu erstellen, die für die Veröffentlichung von Metriken und Protokollen erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbElasticSid",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "EC2Sid",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeVpcEndpoints",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeAvailabilityZones",
                "secretsmanager:ListSecrets"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "KMSSid",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ],
                    "aws:ResourceTag/DocDBElasticFullAccess": "*"
                }
```

```
            }
        },
        {
            "Sid": "KMSGrantSid",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/DocDBElasticFullAccess": "*",
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ]
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        },
        {
            "Sid": "SecretManagerSid",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecretVersionIds",
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:GetResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "secretsmanager:ResourceTag/DocDBElasticFullAccess": "*"
                },
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudwatchSid",
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "SLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDB-ElasticServiceRolePolicy

Sie können nicht `AmazonDocDBElasticServiceRolePolicy` an Ihre AWS Identity and Access Management Entitäten anfügen. Diese Richtlinie ist an eine serviceverknüpfte Rolle angehängt, die es Amazon DocumentDB ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen in elastischen Clustern.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
```

```
                    ]
                }
            }
        }
    ]
}
```

Amazon DocumentDB-Updates für - AWS verwaltete Richtlinien

Änderung	Beschreibung	Datum
AmazonDocDBElasticFullAccess, AmazonDocDBConsoleFullAccess – Änderung	Richtlinien aktualisiert, um Cluster-Start-/Stoppaktionen hinzuzufügen und Cluster-Snapshot-Aktionen zu kopieren.	2/21/2024
AmazonDocDBElasticReadOnlyAccess, AmazonDocDBElasticFullAccess – Änderung	Richtlinien wurden aktualisiert, um eine `cloudwatch:GetMetricData` Aktion hinzuzufügen.	6/21/2023
AmazonDocDBElasticReadOnlyAccess – Neue Richtlinie	Neue verwaltete Richtlinie für elastische Amazon DocumentDB-Cluster	6/8/2023
AmazonDocDBElasticFullAccess – Neue Richtlinie	Neue verwaltete Richtlinie für elastische Amazon DocumentDB-Cluster	6/5/2023
AmazonDocDB-ElasticServiceRolePolicy – Neue Richtlinie.	Amazon DocumentDB erstellt eine neue serviceverknüpfte AWSServiceRoleForDocDB-Elastic-Rolle für elastische Amazon DocumentDB-Cluster	11/30/2022
AmazonDocDBConsoleFullAccess – Änderung	Richtlinie aktualisiert, um globale und elastische Cluster-Berechtigungen für Amazon DocumentDB hinzuzufügen	11/30/2022

Änderung	Beschreibung	Datum
AmazonDocDBConsoleFullAccess, AmazonDocDBFullAccess, AmazonDocDBReadOnlyAccess – Neue Richtlinie	Servicestart	1/19/2017

Amazon DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen

Verwenden Sie die folgenden Abschnitte als Referenz, wenn Sie Berechtigungsrichtlinien einrichten Verwenden von identitätsbasierten Richtlinien (IAM-Richtlinien) für Amazon DocumentDB und schreiben, die Sie einer IAM-Identität anfügen können (identitätsbasierte Richtlinien).

Im Folgenden werden alle Amazon DocumentDB-API-Operationen aufgeführt. In der Liste sind die entsprechenden Aktionen enthalten, für die Sie Berechtigungen zum Ausführen der Aktion erteilen können, die AWS Ressource, für die Sie die Berechtigungen erteilen können, und Bedingungsschlüssel, die Sie für eine differenzierte Zugriffskontrolle einschließen können. Sie geben die Aktionen im Feld `Action` der Richtlinie, den Ressourcenwert im Feld `Resource` der Richtlinie und die Bedingungen im Feld `Condition` der Richtlinie an. Weitere Informationen über Bedingungen finden Sie unter Angeben von Bedingungen in einer Richtlinie.

Sie können AWS-weite Bedingungsschlüssel in Ihren Amazon DocumentDB-Richtlinien verwenden, um Bedingungen auszudrücken. Eine vollständige Liste der AWS-weiten Schlüssel finden Sie unter Verfügbare Schlüssel im IAM-Benutzerhandbuch.

Sie können IAM-Richtlinien mit dem IAM-Richtliniensimulator testen. Es stellt automatisch eine Liste der Ressourcen und Parameter bereit, die für jede AWS Aktion erforderlich sind, einschließlich Amazon DocumentDB-Aktionen. Der IAM-Richtliniensimulator bestimmt die Berechtigungen, die für jede der von Ihnen angegebenen Aktionen erforderlich sind. Informationen zum IAM-Richtliniensimulator finden Sie unter Testen von IAM-Richtlinien mit dem IAM-Richtliniensimulator im IAM-Benutzerhandbuch.

> **Note**
>
> Um eine Aktion anzugeben, verwenden Sie das Präfix `rds:` gefolgt vom Namen der API-Operation (z. B. `rds:CreateDBInstance`).

Im Folgenden werden Amazon-RDS-API-Operationen und die zugehörigen Aktionen, Ressourcen und Bedingungsschlüssel aufgeführt.

Themen

- Amazon DocumentDB-Aktionen, die Berechtigungen auf Ressourcenebene unterstützen
- Amazon DocumentDB-Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen

Amazon DocumentDB-Aktionen, die Berechtigungen auf Ressourcenebene unterstützen

Berechtigungen auf Ressourcenebene bieten die Möglichkeit, die Ressourcen anzugeben, für die Benutzer Aktionen ausführen dürfen. Amazon DocumentDB unterstützt teilweise Berechtigungen auf Ressourcenebene. Das bedeutet, dass Sie bei bestimmten Amazon DocumentDB-Aktionen steuern können, wann Benutzer diese Aktionen verwenden dürfen, basierend auf Bedingungen, die erfüllt sein müssen, oder bestimmten Ressourcen, die Benutzer verwenden dürfen. Beispielsweise können Sie Benutzern die Berechtigung erteilen, nur bestimmte Instances zu ändern.

Im Folgenden werden Amazon DocumentDB-API-Operationen und die zugehörigen Aktionen, Ressourcen und Bedingungsschlüssel aufgeführt.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS geteilt wird.

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
AddTagsToResource	Instance	`rds:db-tag`

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
rds:AddTagsToResource	arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ApplyPendingMaintenanceAction rds:ApplyPendingMaintenanceAction	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
CopyDBClusterSnapshot rds:CopyDBClusterSnapshot	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBCluster rds:CreateDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
CreateDBClusterParameterGroup rds:CreateDBClusterParameterGroup	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
CreateDBClusterSnapshot rds:CreateDBClusterSnapshot	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBInstance rds:CreateDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
CreateDBSubnetGroup rds:CreateDBSubnetGroup	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DeleteDBInstance rds:DeleteDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
DeleteDBSubnetGroup rds:DeleteDBSubnetGroup	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DescribeDBClusterParameterGroups rds:DescribeDBClusterParameterGroups	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
DescribeDBClusterParameters rds:DescribeDBClusterParameters	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
DescribeDBClusters rds:DescribeDBClusters	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
DescribeDBClusterSnapshotAttributes rds:DescribeDBClusterSnapshotAttributes	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
DescribeDBSubnetGroups rds:DescribeDBSubnetGroups	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
DescribePendingMaintenanceActions rds:DescribePendingMaintenanceActions	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
FailoverDBCluster rds:FailoverDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
ListTagsForResource rds:ListTagsForResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ModifyDBCluster rds:ModifyDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
ModifyDBClusterParameterGroup rds:ModifyDBClusterParameterGroup	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
ModifyDBClusterSnapshotAttribute rds:ModifyDBClusterSnapshotAttribute	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
ModifyDBInstance rds:ModifyDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
RebootDBInstance rds:RebootDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
RemoveTagsFromResource rds:RemoveTagsFromResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ResetDBClusterParameterGroup rds:ResetDBClusterParameterGroup	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Amazon DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
RestoreDBClusterFromSnapshot rds:RestoreDBClusterFromSnapshot	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
RestoreDBClusterToPointInTime rds:RestoreDBClusterToPointInTime	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag

Amazon DocumentDB-Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen

Sie können alle Amazon DocumentDB-Aktionen in einer IAM-Richtlinie verwenden, um Benutzern entweder die Berechtigung zur Verwendung dieser Aktion zu erteilen oder zu verweigern. Nicht alle Amazon DocumentDB-Aktionen unterstützen jedoch Berechtigungen auf Ressourcenebene, mit denen Sie die Ressourcen angeben können, für die eine Aktion ausgeführt werden kann. Die folgenden Amazon DocumentDB-API-Aktionen unterstützen derzeit keine Berechtigungen auf Ressourcenebene. Um diese Aktionen in einer IAM-Richtlinie verwenden zu können, müssen Sie

Benutzern daher die Berechtigung erteilen, alle Ressourcen für die Aktion zu verwenden, indem Sie einen * Platzhalter für das -ResourceElement in Ihrer Anweisung verwenden.

- rds:DescribeDBClusterSnapshots
- rds:DescribeDBInstances

Amazon-DocumentDB-Benutzer

In Amazon DocumentDB authentifizieren sich Benutzer bei einem Cluster in Verbindung mit einem Passwort. Jeder Cluster verfügt über primäre Anmeldeinformationen, die bei der Clustererstellung eingerichtet werden.

> **Note**
>
> Allen neuen Benutzern, die vor dem 26. März 2020 erstellt wurden, wurden die dbAdminAnyDatabase-, readWriteAnyDatabase- und clusterAdmin-Rollen erteilt. Es wird empfohlen, alle Benutzer neu zu bewerten und die Rollen nach Bedarf zu ändern, um die geringstmöglichen Berechtigungen für alle Benutzer in Ihren Clustern zu erzwingen. Weitere Informationen finden Sie unter Datenbankzugriff mit rollenbasierter Zugriffskontrolle.

Primär und**serviceadmin** Benutzer

Ein neu erstellter Amazon DocumentDB-Cluster hat zwei Benutzer: den Hauptbenutzer und denserviceadmin Benutzer.

Der Hauptbenutzer ist ein einzelner, privilegierter Benutzer, der administrative Aufgaben ausführen und zusätzliche Benutzer mit Rollen erstellen kann. Wenn Sie zum ersten Mal eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen, müssen Sie sich mit den primären Anmeldeinformationen authentifizieren. Der Hauptbenutzer erhält diese Administratorberechtigungen für einen Amazon DocumentDB-Cluster, wenn dieser Cluster erstellt wird, und ihm wird die Rolle von zugewiesenroot.

Der serviceadmin-Benutzer wird implizit beim Erstellen des Clusters erstellt. Jeder Amazon DocumentDB-Cluster hat einenserviceadmin Benutzer,AWS der Ihnen die Möglichkeit bietet, Ihren Cluster zu verwalten. Sie können sich mit serviceadmin nicht anmelden, den Benutzer löschen oder umbenennen, sein Passwort ändern oder die Berechtigungen ändern. Jeder Versuch, das zu tun, führt zu einem Fehler.

> **Note**
>
> Die primären`serviceadmin` Benutzer und die Benutzer für einen Amazon DocumentDB-Cluster können nicht gelöscht werden, und die Rolle von`root` für den Hauptbenutzer kann nicht aufgehoben werden.
> Wenn Sie Ihr Hauptbenutzer-Passwort haben Sie es mit der die die die die die dieAWS Management Console die dieAWS CLI.

Erstellen weiterer Benutzer

Nachdem Sie sich als primärer Benutzer (oder als beliebiger Benutzer mit dieser Rolle`createUser`) verbunden haben, können Sie wie unten gezeigt einen neuen Benutzer erstellen.

```
db.createUser(
    {
        user: "sample-user-1",
        pwd: "password123",
        roles:
            [{"db":"admin", "role":"dbAdminAnyDatabase" }]
    }
)
```

Um Benutzerdetails anzuzeigen, können Sie den Befehl `show users` wie folgt verwenden. Sie können Benutzer zusätzlich mit dem Befehl `dropUser` entfernen. Weitere Informationen finden Sie unter Allgemeine Befehle.

```
show users
{
    "_id" : "serviceadmin",
    "user" : "serviceadmin",
    "db" : "admin",
    "roles" : [
     {
            "role" : "root",
            "db" : "admin"
        }
    ]
},
```

```
{
    "_id" : "myPrimaryUser",
    "user" : "myPrimaryUser",
    "db" : "admin",
    "roles" : [
     {
            "role" : "root",
            "db" : "admin"
        }
    ]
},

{
    "_id" : "sample-user-1",
    "user" : "sample-user-1",
    "db" : "admin",
    "roles" : [
     {
            "role" : "dbAdminAnyDatabase",
            "db" : "admin"
     }
    ]
}
```

Im obigen Beispiel wird der neue Benutzer `sample-user-1` der `admin`-Datenbank zugewiesen. Dies ist bei einem neuen Benutzer immer der Fall. Amazon DocumentDB hat nicht das Konzept einer`authenticationDatabase` und daher wird die gesamte Authentifizierung im Kontext der`admin` Datenbank durchgeführt.

Wenn Sie beim Erstellen von Benutzern dasdb Feld bei der Angabe der Rolle weglassen, weist Amazon DocumentDB die Rolle implizit der Datenbank zu, für die die Verbindung hergestellt wird. Wenn Ihre Verbindung beispielsweise für die Datenbank `sample-database` hergestellt wird und Sie den folgenden Befehl ausführen, wird der Benutzer `sample-user-2` in der Datenbank `admin` erstellt und verfügt über `readWrite`-Berechtigungen für die Datenbank `sample-database`.

```
db.createUser(
    {
        user: "sample-user-2",
        pwd: "password123",
        roles:
```

```
            ["readWrite"]
    }
)
```

Wenn Sie Benutzer mit Rollen erstellen, die über alle Datenbanken hinweg erfasst sind (z. B. `readInAnyDatabase`), müssen Sie sich beim Erstellen des Benutzers entweder im Kontext der `admin`-Datenbank befinden oder beim Erstellen des Benutzers explizit die Datenbank für die Rolle angeben.

Um den Kontext Ihrer Datenbank zu wechseln, können Sie den folgenden Befehl verwenden.

```
use admin
```

Weitere Informationen zur rollenbasierten Zugriffssteuerung und zum Erzwingen geringstmöglicher Berechtigungen unter den Benutzern im Cluster finden Sie unter Datenbankzugriff mit rollenbasierter Zugriffskontrolle.

Automatisch rotierende Passwörter für Amazon DocumentDB

Mit AWS Secrets Manager können Sie fest codierte Anmeldeinformationen im Code (einschließlich Passwörter) durch einen API-Aufruf an Secrets Manager ersetzen und das Secret programmgesteuert abrufen. Dadurch wird sichergestellt, dass das Secret nicht kompromittiert werden kann, wenn jemand Ihren Code durchsucht, da es sich gar nicht dort befindet. Außerdem können Sie Secrets Manager so konfigurieren, dass er das Secret automatisch nach einem von Ihnen festgelegten Zeitplan rotiert. So können Sie Secrets mit langer Einsatzdauer durch Secrets mit kurzer Einsatzdauer ersetzen und damit das Risiko einer Kompromittierung erheblich verringern.

Mit Secrets Manager können Sie Ihre Passwörter für Amazon DocumentDB (d. h. geheimen Schlüssel) automatisch mit einerAWS Lambda -Funktion von Secrets Manager rotieren lassen.

Weitere Informationen zur nativen Integration mit Amazon DocumentDBAWS Secrets Manager und der nativen Integration mit Amazon DocumentDB finden Sie hier:

- Blog: So wechseln Sie die Anmeldeinformationen für Amazon DocumentDB und Amazon Redshift inAWS Secrets Manager
- Was istAWS Secrets Manager?
- Rotieren von geheimen Schlüsseln für Amazon DocumentDB

Datenbankzugriff mit rollenbasierter Zugriffskontrolle

Sie können den Zugriff auf die Aktionen einschränken, die Benutzer mithilfe der rollenbasierten Zugriffskontrolle (RBAC) in Amazon DocumentDB (mit MongoDB-Kompatibilität) für Datenbanken ausführen können. Amazon DocumentDB MongoDB Bei RBAC werden einem Benutzer eine oder mehrere Rollen gewährt. Diese Rollen bestimmen die Operationen, die ein Benutzer für Datenbankressourcen ausführen kann. Amazon DocumentDB unterstützt derzeit sowohl integrierte Rollen, die auf Datenbankebene ausgerichtet sind, wie `read`, `readWrite`, `readAnyDatabaseclusterAdmin`, und benutzerdefinierte Rollen, die auf bestimmte Aktionen und detaillierte Ressourcen wie Sammlungen basierend auf Ihren Anforderungen beschränkt werden können.

Zu den häufigsten Anwendungsfällen für RBAC gehören die Durchsetzung von geringsten Rechten, indem Benutzer mit schreibgeschütztem Zugriff auf die Datenbanken oder Sammlungen in einem Cluster erstellt werden, und Anwendungsdesigns mit mehreren Mandanten, die es einem einzelnen Benutzer ermöglichen, auf eine bestimmte Datenbank oder Sammlung in einem Cluster zuzugreifen.

> **Note**
>
> Allen neuen Benutzern, die vor dem 26. März 2020 erstellt wurden, wurden die `dbAdminAnyDatabase`-, `readWriteAnyDatabase`- und `clusterAdmin`-Rollen erteilt. Es wird empfohlen, alle vorhandenen Benutzer neu zu bewerten und die Rollen nach Bedarf zu ändern, um die geringstmöglichen Berechtigungen für Ihre Cluster zu erzwingen.

Themen

RBAC-Konzepte

Im Folgenden finden Sie wichtige Begriffe und Konzepte zur rollenbasierten Zugriffssteuerung. Weitere Informationen zu Amazon DocumentDB-Benutzern finden Sie unter Amazon-DocumentDB-Benutzer.

- Benutzer – Eine einzelne Entität, die sich bei der Datenbank authentifizieren und Vorgänge ausführen kann.
- Passwort – Ein Secret, das zur Authentifizierung des Benutzers verwendet wird.
- Rolle – Autorisiert einen Benutzer zum Ausführen von Aktionen für eine oder mehrere Datenbanken.
- Admin Database – Die Datenbank, in der Benutzer gespeichert und autorisiert werden.
- Datenbank (**db**) – Der Namespace innerhalb von Clustern, der Sammlungen zum Speichern von Dokumenten enthält.

Der folgende Befehl erstellt einen Benutzer mit Namen `sample-user`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

In diesem Beispiel:

- `user: "sample-user"` – Gibt den Benutzernamen an.
- `pwd: "abc123"` – Gibt das Benutzerpasswort an.
- `role: "read", "db: "sample-database"` – Zeigt an, dass der Benutzer über Leseberechtigungen in `sample-user` verfügt`sample-database`.

Das folgende Beispiel zeigt die Ausgabe, nachdem Sie den Benutzer `sample-user` mit `db.getUser(sample-user)` erhalten. In diesem Beispiel befindet sich der Benutzer `sample-user` in der Datenbank `admin`, besitzt jedoch die „read“-Rolle für die Datenbank `sample-database`.

Wenn Sie beim Erstellen von Benutzern das -dbFeld weglassen, wenn Sie die Rolle angeben, wird Amazon DocumentDB die Rolle implizit der Datenbank zuordnen, für die die Verbindung ausgestellt wird. Wenn Ihre Verbindung beispielsweise für die Datenbank `sample-database` hergestellt wird und Sie den folgenden Befehl ausführen, wird der Benutzer `sample-user` in der Datenbank `admin` erstellt und verfügt über `readWrite`-Berechtigungen für die Datenbank `sample-database`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: ["readWrite"]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"sample-user",
   "roles":[
      {
         "db":"sample-database",
         "role":"readWrite"
      }
   ]
}
```

Wenn Sie Benutzer mit Rollen erstellen, die über alle Datenbanken hinweg erfasst sind (z. B. `readAnyDatabase`), müssen Sie sich beim Erstellen des Benutzers entweder im Kontext der Datenbank `admin` befinden oder beim Erstellen des Benutzers explizit die Datenbank für die Rolle angeben. Um Befehle für die Datenbank `admin` auszugeben, können Sie den Befehl `use admin` verwenden. Weitere Informationen finden Sie unter Allgemeine Befehle.

Erste Schritte mit integrierten RBAC-Rollen

Um Ihnen den Einstieg in die rollenbasierte Zugriffssteuerung zu erleichtern, führen Sie in diesem Abschnitt ein Beispielszenario zum Erzwingen der geringstmöglichen Berechtigungen durch das Erstellen von Rollen für drei Benutzer mit unterschiedlichen Auftragsfunktionen durch.

- `user1` ist ein neuer Manager, der in der Lage sein muss, alle Datenbanken in einem Cluster anzuzeigen und darauf zuzugreifen.
- `user2` ist ein neuer Mitarbeiter, der Zugriff auf nur eine Datenbank, `sample-database-1`, in demselben Cluster benötigt.
- `user3` ist ein vorhandener Mitarbeiter, der im selben Cluster eine andere Datenbank, `sample-database-2`, anzeigen und darauf zugreifen muss, auf die er vorher keinen Zugriff hatten.

Zu einem späteren Zeitpunkt verlassen sowohl `user1` als auch `user2` das Unternehmen, so dass ihr Zugang widerrufen werden muss.

Um Benutzer zu erstellen und Rollen zu erteilen, muss der Benutzer, mit dem Sie sich beim Cluster authentifizieren, über eine zugeordnete Rolle verfügen, die Aktionen für `createUser` und `grantRole` ausführen kann. Zum Beispiel können die Rollen `admin` und `userAdminAnyDatabase` beide solche Fähigkeiten gewähren. Informationen zu Aktionen pro Rolle finden Sie unter Datenbankzugriff mit rollenbasierter Zugriffskontrolle.

> **Note**
>
> In Amazon DocumentDB werden alle Benutzer- und Rollenoperationen (z. B. `create`, `get`, `drop`, `grant`, usw.) implizit in der `admin` Datenbank ausgeführt`revoke`, unabhängig davon, ob Sie Befehle für die `admin` Datenbank ausgeben oder nicht.

Um zunächst zu verstehen, was die aktuellen Benutzer und Rollen im Cluster sind, können Sie den Befehl `show users` ausführen, wie im folgenden Beispiel. Sie sehen zwei Benutzer, den `serviceadmin`- und den Master-Benutzer für den Cluster. Diese beiden Benutzer sind immer vorhanden und können nicht gelöscht werden. Weitere Informationen finden Sie unter Amazon-DocumentDB-Benutzer.

```
show users
```

Erstellen Sie für `user1` eine Rolle mit Lese- und Schreibzugriff auf alle Datenbanken im gesamten Cluster mit dem folgenden Befehl.

```
db.createUser({user: "user1", pwd: "abc123", roles: [{role: "readWriteAnyDatabase", db:
 "admin"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Erstellen Sie für `user2` eine Rolle mit schreibgeschütztem Zugriff auf die Datenbank `sample-database-1` mit dem folgenden Befehl.

```
db.createUser({user: "user2", pwd: "abc123", roles: [{role: "read", db: "sample-
database-1"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user2",
   "roles":[
      {
         "role":"read",
         "db":"sample-database-1"
      }
   ]
}
```

Um das Szenario zu simulieren, dass `user3` ein vorhandener Benutzer ist, erstellen Sie zuerst den Benutzer `user3` und weisen Sie dann `user3` eine neue Rolle zu.

```
db.createUser({user: "user3", pwd: "abc123", roles: [{role: "readWrite", db: "sample-
database-1"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user3",
   "roles":[
      {
         "role":"readWrite",
         "db":"sample-database-1"
      }
   ]
}
```

Nachdem der Benutzer `user3` erstellt wurde, weisen Sie `user3` die `read`-Rolle für `sample-database-2` zu.

```
db.grantRolesToUser("user3", [{role: "read", db: "sample-database-2"}])
```

Schließlich verlassen sowohl `user1` als auch `user2` das Unternehmen und ihr Zugriff auf den Cluster muss widerrufen werden. Sie können dies tun, indem Sie die Benutzer wie folgt löschen.

```
db.dropUser("user1")
db.dropUser("user2")
```

Um sicherzustellen, dass alle Benutzer über die entsprechenden Rollen verfügen, können Sie alle Benutzer mit dem folgenden Befehl auflisten.

```
show users
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "_id":"serviceadmin",
   "user":"serviceadmin",
   "db":"admin",
   "roles":[
```

```
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"master-user",
   "user":"master-user",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"user3",
   "user":"user3",
   "db":"admin",
   "roles":[
      {
         "db":"sample-database-2",
         "role":"read"
      },
      {
         "db":"sample-database-1",
         "role":"readWrite"
      }
   ]
}
```

Erste Schritte mit benutzerdefinierten RBAC-Rollen

Um Ihnen den Einstieg in benutzerdefinierte Rollen zu erleichtern, führt Sie dieser Abschnitt durch ein Beispielszenario zur Durchsetzung der geringsten Berechtigungen, indem Rollen für drei Benutzer mit unterschiedlichen Auftragsfunktionen erstellt werden.

In diesem Beispiel gilt Folgendes:

- `user1` ist ein neuer Manager, der in der Lage sein muss, alle Datenbanken in einem Cluster anzuzeigen und darauf zuzugreifen.

- `user2` ist ein neuer Mitarbeiter, der nur die Aktion „Suchen“ für eine Datenbank, `sample-database-1`, in demselben Cluster benötigt.
- `user3` ist ein vorhandener Mitarbeiter, der eine bestimmte Sammlung, col2, in einer anderen Datenbank anzeigen und darauf zugreifen muss, auf `sample-database-2` die er zuvor keinen Zugriff hatte, im selben Cluster.
- Erstellen Sie für `user1` eine Rolle mit Lese- und Schreibzugriff auf alle Datenbanken im gesamten Cluster mit dem folgenden Befehl.

```
db.createUser(
 {
     user: "user1", pwd: "abc123",
     roles: [{role: "readWriteAnyDatabase", db: "admin"}]
 }
 )
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

`user2`Erstellen Sie für mit `sample-database-1` dem folgenden Befehl eine Rolle mit „Find“-Berechtigungen für alle Sammlungen in der Datenbank. Beachten Sie, dass diese Rolle sicherstellen würde, dass alle zugehörigen Benutzer nur Suchabfragen ausführen können.

```
db.createRole(
{
    role: "findRole",
    privileges: [
    {
        resource: {db: "sample-database-1", collection: ""}, actions: ["find"]
    }],
    roles: []
```

```
}
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"findRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "find"
         ]
      }
   ],
   "roles":[

   ]
}
```

Erstellen Sie als Nächstes den Benutzer (`user2`) und fügen Sie die kürzlich erstellte Rolle an den Benutzer `findRole` an.

```
db.createUser(
{
    user: "user2",
    pwd: "abc123",
    roles: []
})

 db.grantRolesToUser("user2",["findRole"])
```

Um das Szenario zu simulieren, bei dem es sich um einen vorhandenen Benutzer `user3` handelt, erstellen Sie zunächst den Benutzer `user3`und dann eine neue Rolle namens collectionRole, die wir im nächsten Schritt auf festlegen werden`user3`.

Jetzt können Sie eine neue Rolle zuweisen`user3`. Diese neue Rolle ermöglicht es `user3` , eine bestimmte Sammlungsspalte in einzufügen, zu aktualisieren, zu löschen und den Zugriff auf eine bestimmte Sammlungsspalte zu finden`sample-database-2`.

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Nachdem der Benutzer erstellt `user3` wurde, können Sie `user3` der Rolle gewährencollectionFind.

```
db.grantRolesToUser("user3",["collectionRole"])
```

Schließlich verlassen sowohl `user1` als auch `user2` das Unternehmen und ihr Zugriff auf den Cluster muss widerrufen werden. Sie können dies tun, indem Sie die Benutzer wie folgt löschen.

```
db.dropUser("user1")
db.dropUser("user2")
```

Um sicherzustellen, dass alle Benutzer über die entsprechenden Rollen verfügen, können Sie alle Benutzer mit dem folgenden Befehl auflisten.

```
show users
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "_id":"serviceadmin",
   "user":"serviceadmin",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"master-user",
   "user":"master-user",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
```

```
   "_id":"user3",
   "user":"user3",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"collectionRole"
      }
   ]
}
```

Herstellen einer Verbindung mit Amazon DocumentDB als Benutzer

Wenn Sie eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen, stellen Sie eine Verbindung im Kontext einer bestimmten Datenbank her. Wenn Sie in der Verbindungszeichenfolge keine Datenbank angeben, werden Sie standardmäßig automatisch im Kontext der Datenbank `test` mit dem Cluster verbunden. Alle Befehle auf Sammlungsebene wie `insert` und `find` werden für Sammlungen in der Datenbank `test` ausgegeben.

Um die Datenbank zu sehen, für die Sie sich im Kontext von oder befinden – mit anderen Worten – die Befehle ausgibt – verwenden Sie den db Befehl in der mongo-Shell wie folgt.

Abfrage:

```
db
```

Ausgabe:

```
test
```

Obwohl sich die Standardverbindung möglicherweise im Kontext der Datenbank `test` befindet, bedeutet dies nicht unbedingt, dass der Benutzer, der der Verbindung zugeordnet ist, berechtigt ist, Aktionen für die Datenbank `test` auszuführen. Wenn Sie sich im vorangegangenen Beispielszenario als Benutzer `user3` authentifizieren, der die `readWrite`-Rolle für die Datenbank `sample-database-1` besitzt, ist der Standardkontext Ihrer Verbindung die Datenbank `test`. Wenn Sie jedoch versuchen, ein Dokument in eine Sammlung in der Datenbank `test` einzufügen, erhalten Sie die Fehlermeldung Autorisierungsfehler. Dies liegt daran, dass dieser Benutzer nicht berechtigt ist, diesen Befehl in dieser Datenbank auszuführen, wie unten gezeigt.

Abfrage:

```
db
```

Ausgabe:

```
test
```

Abfrage:

```
db.col.insert({x:1})
```

Ausgabe:

```
WriteCommandError({ "ok" : 0, "code" : 13, "errmsg" : "Authorization failure" })
```

Wenn Sie den Kontext Ihrer Verbindung zur Datenbank `sample-database-1` ändern, können Sie in die Sammlung schreiben, für die der Benutzer die Berechtigung dazu hat.

Abfrage:

```
use sample-database-1
```

Ausgabe:

```
switched to db sample-database-1
```

Abfrage:

```
db.col.insert({x:1})
```

Ausgabe:

```
WriteResult({ "nInserted" : 1})
```

Wenn Sie sich bei einem Cluster mit einem bestimmten Benutzer authentifizieren, können Sie die Datenbank auch in der Verbindungszeichenfolge angeben. Dadurch entfällt die Notwendigkeit, den Befehl `use` auszuführen, nachdem der Benutzer in der Datenbank `admin` authentifiziert wurde.

Die folgende Verbindungszeichenfolge authentifiziert den Benutzer für die Datenbank `admin`, aber der Kontext der Verbindung wird für die Datenbank `sample-database-1` verwendet.

```
mongo "mongodb://user3:abc123@sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
sample-database-2"
```

Allgemeine Befehle

Dieser Abschnitt enthält Beispiele für gängige Befehle unter Verwendung der rollenbasierten Zugriffskontrolle in Amazon DocumentDB . Sie müssen sich im Kontext der Datenbank `admin` befinden, um Benutzer und Rollen zu erstellen und zu ändern. Sie können den Befehl `use admin` verwenden, um zur Datenbank `admin` zu wechseln.

> **Note**
>
> Änderungen an den Benutzern und Rollen erfolgen implizit in der Datenbank `admin`. Wenn Sie Benutzer mit Rollen erstellen, die über alle Datenbanken hinweg erfasst sind (z. B. `readAnyDatabase`), müssen Sie sich beim Erstellen des Benutzers entweder im Kontext der Datenbank `admin` befinden (d. h. `use admin`) oder beim Erstellen des Benutzers explizit die Datenbank für die Rolle angeben (wie in Beispiel 2 in diesem Abschnitt gezeigt).

Beispiel 1: Erstellen Sie einen Benutzer mit der `read` Rolle für die Datenbank `foo`.

```
db.createUser({user: "readInFooBar", pwd: "abc123", roles: [{role: "read", db:
 "foo"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"readInFooBar",
   "roles":[
      {
         "role":"read",
         "db":"foo"
      }
   ]
}
```

Beispiel 2: Erstellen Sie einen Benutzer mit Lesezugriff für alle Datenbanken.

```
db.createUser({user: "readAllDBs", pwd: "abc123", roles: [{role: "readAnyDatabase", db:
 "admin"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"readAllDBs",
   "roles":[
      {
         "role":"readAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Beispiel 3: Erteilen einer `read` Rolle für einen vorhandenen Benutzer in einer neuen Datenbank.

```
db.grantRolesToUser("readInFooBar", [{role: "read", db: "bar"}])
```

Beispiel 4: Aktualisieren der Rolle eines Benutzers.

```
db.updateUser("readInFooBar", {roles: [{role: "read", db: "foo"}, {role: "read", db:
 "baz"}]})
```

Beispiel 5: Widerrufen des Zugriffs auf eine Datenbank für einen Benutzer.

```
db.revokeRolesFromUser("readInFooBar", [{role: "read", db: "baz"}])
```

Beispiel 6: Beschreiben einer integrierten Rolle.

```
db.getRole("read", {showPrivileges:true})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"read",
   "db":"sample-database-1",
   "isBuiltin":true,
   "roles":[
```

```
   ],
   "inheritedRoles":[

   ],
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "changeStream",
            "collStats",
            "dbStats",
            "find",
            "killCursors",
            "listCollections",
            "listIndexes"
         ]
      }
   ],
   "inheritedPrivileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "changeStream",
            "collStats",
            "dbStats",
            "find",
            "killCursors",
            "listCollections",
            "listIndexes"
         ]
      }
   }
```

Beispiel 7: Löschen eines Benutzers aus dem Cluster.

```
db.dropUser("readInFooBar")
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
true
```

Beispiel 8: Erstellen einer Rolle mit Lese- und Schreibzugriff auf eine bestimmte Sammlung

```
db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Beispiel 9: Erstellen eines Benutzers und Zuweisen einer benutzerdefinierten Rolle

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.grantRolesToUser("user3",["collectionRole"])
```

Beispiel 10: Erteilen zusätzlicher Berechtigungen für eine benutzerdefinierte Rolle

```
db.grantPrivilegesToRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col1" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Beispiel 11: Entfernen von Berechtigungen aus einer benutzerdefinierten Rolle

```
db.revokePrivilegesFromRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col2" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Beispiel 12: Aktualisieren einer vorhandenen benutzerdefinierten Rolle

```
db.updateRole(
  "collectionRole",
  {
    privileges: [
    {
        resource: {db: "sample-database-3", collection: "sample-collection-3"},
 actions: ["find", "update", "insert", "remove"]
```

```
    }],
    roles: []
  }
)
```

Funktionsunterschiede

In Amazon DocumentDB werden Benutzer- und Rollendefinitionen in der `admin` Datenbank gespeichert und Benutzer werden anhand der `admin` Datenbank authentifiziert. Diese Funktionalität unterscheidet sich von der MongoDB Community Edition, ist aber konsistent mit MongoDB Atlas.

Amazon DocumentDB unterstützt auch Änderungsstreams, die eine zeitlich geordnete Abfolge von Änderungsereignissen bereitstellen, die innerhalb der Sammlungen Ihres Clusters auftreten. Die `listChangeStreams` Aktion wird auf Clusterebene (d. h. auf alle Datenbanken) angewendet, und die `modifyChangeStreams` Aktion kann auf Datenbankebene und Clusterebene angewendet werden.

Einschränkungen

Die folgende Tabelle enthält die Limits für die rollenbasierte Zugriffskontrolle in Amazon DocumentDB .

Beschreibung	Limit
Anzahl Benutzer pro Cluster	1000
Anzahl der Rollen, die einem Benutzer zugeordnet sind	1000
Anzahl der benutzerdefinierten Rollen	100
Anzahl der Ressourcen, die einer Berechtigung zugeordnet sind	100

Datenbankzugriff mit rollenbasierter Zugriffskontrolle

Mit der rollenbasierten Zugriffssteuerung können Sie einen Benutzer erstellen und ihm eine oder mehrere Rollen erteilen, um zu bestimmen, welche Operationen der Benutzer in einer Datenbank oder einem Cluster ausführen kann.

Im Folgenden finden Sie eine Liste der integrierten Rollen, die derzeit in Amazon DocumentDB unterstützt werden.

> **Note**
>
> In Amazon DocumentDB 4.0 und 5.0 können die `ListDatabase` Befehle `ListCollection` und optional die `authorizedDatabases` Parameter `authorizedCollections` und verwenden, um die Sammlungen und Datenbanken aufzulisten, auf die der Benutzer zugreifen darf, wobei die `listDatabase` Rollen `listCollections` und erforderlich sind. Außerdem haben Benutzer jetzt die Möglichkeit, ihre eigenen Cursor zu beenden, ohne dass die `KillCursor` Rolle erforderlich ist.

Database user

Rollenname	Beschreibung	Aktionen

`read`	Gewährt einem Benutzer Lesezugriff auf die angegebene Datenbank.	`changeStreams`
		`collStats`
		`dbStats`
		`find`
		`killCursors`
		`listIndexes`
		`listCollections`

`readWrite`	Gewährt dem Benutzer Lese- und Schreibzugriff auf die angegebene Datenbank.	Alle Aktionen von `read`-Berechtigungen.
		`createCollection`
		`dropCollection`

Rollenname	Beschreibung	Aktionen

		`createIndex`
		`dropIndex`
		`insert`
		`killCursors`
		`listIndexes`
		`listCollections`
		`remove`
		`update`

Cluster user

Rollenname	Beschreibung	Aktionen

readAnyDatabase	Gewährt einem Benutzer Lesezugriff auf alle Datenbanken im Cluster.	Alle Aktionen von `read`-Berechtigungen.
		`listChangeStreams`
		`listDatabases`

`readWriteAnyDatabase`	Gewährt einem Benutzer Lese- und Schreibzugriff auf alle Datenbanken im Cluster.	Alle Aktionen von `readWrite` -Berechtigungen.
		`listChangeStreams`
		`listDatabases`

Rollenname	Beschreibung	Aktionen

`userAdminAnyDatabase`	Gewährt einem Benutzer die Möglichkeit, die Rollen oder Berechtigungen, die ein Benutzer hat, der angegebenen Datenbank zuzuweisen und zu ändern.	`changeCustomData`
		`changePassword`
		`createUser`
		`dropRole`
		`dropUser`
		`grantRole`
		`listDatabases`
		`revokeRole`
		`viewRole`
		`viewUser`

`dbAdminAnyDatabase`	Gewährt einem Benutzer die Möglichkeit, Datenbankverwaltungsrollen für jede angegebene Datenbank auszuführen.	Alle Aktionen von `dbAdmin`-Berechtigungen.
		`dropCollection`
		`listDatabases`
		`listChangeStreams`
		`modifyChangeStreams`

Superuser

Rollenname	Beschreibung	Aktionen
`root`	Gewährt einem Benutzer Zugriff auf die Ressourcen und Operationen aller folgenden Rollen zusammen: `readWriteAnyDatabase` , `dbAdminAnyDatabase` , `userAdminAnyDatabase` , `clusterAdmin` , `restore` und `backup`.	Alle Aktionen von `readWriteAnyDatabase` , `dbAdminAnyDatabase` , `userAdminAnyDatabase` , `clusterAdmin` , `restore` und `backup`.

Database administrator

Rollenname	Beschreibung	Aktionen

`dbAdmin`	Gewährt einem Benutzer die Möglichkeit, administrative Aufgaben für die angegebene Datenbank auszuführen.	`collMod`
		`collStats`
		`createCollection`
		`createIndex`
		`dropCollection`
		`dropDatabase`
		`dropIndex`
		`dbStats`
		`find`
		`killCursors`

Rollenname	Beschreibung	Aktionen

		`listIndexes`
		
		`listColle ctions`
		
		`modifyCha ngeStreams`

`db0wner`	Gewährt einem Benutzer die Möglichkeit, alle administrativen Aufgaben für die angegeben en Datenbank auszuführen, indem die Rollen `dbAdmin` und `readWrite` kombiniert werden.	Alle Aktionen von `dbAdmin` und `readWrite` .

Logkolliert liert lierung in Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) stellt eine Vielzahl von CloudWatch Amazon--Metriken bereit, die Sie überwachen können, um den Zustand und die Leistung Ihrer Amazon-Document-DB-Cluster und -Instances zu überwachen. Sie können Amazon DocumentDB DocumentDB-Metriken mithilfe verschiedener Tools anzeigen, darunter die Amazon DocumentDB CloudWatch DocumentDB-KonsoleAWS CLI, die Amazon-Konsole und die CloudWatch API. Weitere Informationen zur Überwachung finden Sie unter Amazon-Documententententent.

Zusätzlich zu den CloudWatch Amazon-Metriken können Sie den Profiler verwenden, um die Ausführungszeit und Details der Operationen zu protokollieren, die in Ihrem Cluster ausgeführt wurden. Profiler ist nützlich für die Überwachung der langsamsten Operationen in Ihrem Cluster, um die Leistung einzelner Abfragen und die allgemeine Cluster-Leistung zu verbessern. Wenn diese Option aktiviert ist, werden die Vorgänge in Amazon CloudWatch Logs protokolliert, und Sie können CloudWatch Insight verwenden, um Ihre Amazon DocumentDB DocumentDB-Profildaten zu analysieren, zu überwachen und zu archivieren. Weitere Informationen finden Sie unter Profilierung von Amazon DocumentDB-Vorgängen.

Amazon DocumentDB ist in integriert. Dieser Service zeichnet die Aktionen von Benutzern, Rollen oder einesAWS -Service in Amazon DocumentDB auf.AWS CloudTrail CloudTrail erfasst alleAWS CLI API-Aufrufe für Amazon DocumentDB als Ereignisse, einschließlich Aufrufen von Amazon

DocumentDBAWS Management Console und von Code-Aufrufen an das Amazon DocumentDB-SDK. Weitere Informationen finden Sie unter ProtokolDB-API-API-API-API-API-APIAWS CloudTrail.

Mit Amazon DocumentDB können Sie Ereignisse überprüfen, die in Ihrem Cluster durchgeführt wurden. Beispiele für protokollierte Ereignisse sind erfolgreiche und fehlgeschlagene Authentifizierungsversuche, Drop-Ereignisse für Sammlungen in einer Datenbank oder das Erstellen eines Index. Standardmäßig ist das Auditing in Amazon DocumentDB deaktiviert und erfordert, dass Sie sich für diese Funktion anmelden. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.

Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate

Themen

Das Zertifikat der Zertifizierungsstelle (CA) für Amazon DocumentDB-Cluster wird ab August 2024 aktualisiert. Wenn Sie Amazon DocumentDB-Cluster mit aktiviertem Transport Layer Security (TLS) verwenden (Standardeinstellung) und Sie Ihre Client-Anwendungs- und Serverzertifikate nicht rotiert haben, sind die folgenden Schritte erforderlich, um Verbindungsprobleme zwischen Ihrer Anwendung und Ihren Amazon DocumentDB-Clustern zu beheben.

Die CA- und Serverzertifikate wurden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Client-Anwendungen müssen die neuen CA-Zertifikate zu ihren Trust Stores hinzufügen, und bestehende Amazon DocumentDB DocumentDB-Instances müssen aktualisiert werden, sodass sie die neuen CA-Zertifikate vor diesem Ablaufdatum verwenden können.

Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters

Führen Sie die Schritte in diesem Abschnitt aus, um das CA-Zertifikatspaket Ihrer Anwendung (Schritt 1) und die Serverzertifikate Ihres Clusters (Schritt 2) zu aktualisieren. Bevor Sie die

Änderungen auf Ihre Produktionsumgebungen anwenden, empfehlen wir dringend, diese Schritte in einer Entwicklungs- oder Stagingumgebung zu testen.

> **Note**
>
> Sie müssen die Schritte 1 und 2 jeweils AWS-Region ausführen, wenn Sie Amazon DocumentDB-Cluster haben.

Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung

Laden Sie das neue CA-Zertifikat herunter und aktualisieren Sie Ihre Anwendung so, dass sie das neue CA-Zertifikat verwendet, um TLS-Verbindungen zu Amazon DocumentDB herzustellen. Laden Sie das neue CA-Zertifikatpaket von https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem herunter. Diese Operation lädt die Datei mit dem Namen `global-bundle.pem` herunter.

> **Note**
>
> Wenn Sie auf den Keystore zugreifen, der sowohl das alte CA-Zertifikat (`rds-ca-2019-root.pem`) als auch die neuen CA-Zertifikate (`rds-ca-rsa2048-g1, rds-ca-rsa4096-g1`) enthält, stellen Sie sicher, dass der Keystore auswählt. `global-bundle`

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Aktualisieren Sie als Nächstes Ihre Anwendungen, um das neue Zertifikatpaket zu verwenden. Das neue CA-Paket enthält sowohl das alte CA-Zertifikat (rds-ca-2019) als auch die neuen CA-Zertifikate (2048-g1, 4096-g1). rds-ca-rsa rds-ca-rsa Wenn beide CA-Zertifikate im neuen CA-Paket enthalten sind, können Sie Ihre Anwendung und Ihren Cluster in zwei Schritten aktualisieren.

Informationen dazu, wie Sie überprüfen, ob Ihre Anwendung das neueste CA-Zertifikatspaket verwendet, finden Sie unter Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?. Wenn Sie das neueste CA-Zertifikatspaket in Ihrer Anwendung bereits verwenden, können Sie mit Schritt 2 fortfahren.

Beispiele für die Verwendung eines Zertifizierungsstellenpakets mit Ihrer Anwendung finden Sie unter Datenverschlüsselung während der Übertragung und Verbindung bei aktiviertem TLS herstellen.

> **Note**
>
> Zurzeit akzeptiert der MongoDB Go Driver 1.2.1 nur ein einzelnes CA-Serverzertifikat in `sslcertificateauthorityfile`. Informationen dazu, wie Sie bei aktiviertem TLS eine Verbindung zu Amazon DocumentDB über Go herstellen, finden Sie unter Verbindung bei aktiviertem TLS herstellen.

Schritt 2: Aktualisieren des Serverzertifikats

Nachdem die Anwendung für die Verwendung des neuen CA-Bundles aktualisiert wurde, besteht der nächste Schritt darin, das Serverzertifikat zu aktualisieren, indem jede Instance in einem Amazon DocumentDB-Cluster geändert wird. Informationen zum Ändern von Instances zur Verwendung des neuen Serverzertifikats finden Sie in den folgenden Anweisungen.

Amazon DocumentDB stellt die folgenden Zertifizierungsstellen bereit, um das DB-Serverzertifikat für eine DB-Instance zu signieren:

- rds-ca-rsa2048-g1 — Verwendet in den meisten Regionen eine Zertifizierungsstelle mit dem RSA 2048-Algorithmus für private Schlüssel und dem SHA256-Signaturalgorithmus. AWS Diese CA unterstützt die automatische Rotation von Serverzertifikaten.
- rds-ca-rsa4096-g1 — Verwendet eine Zertifizierungsstelle mit dem RSA 4096-Algorithmus für private Schlüssel und dem SHA384-Signaturalgorithmus. Diese CA unterstützt die automatische Rotation von Serverzertifikaten.

> **Note**
>
> Wenn Sie die AWS CLI verwenden, können Sie die Gültigkeitsdauer der oben aufgeführten Zertifizierungsstellen mithilfe von describe-certificates überprüfen.

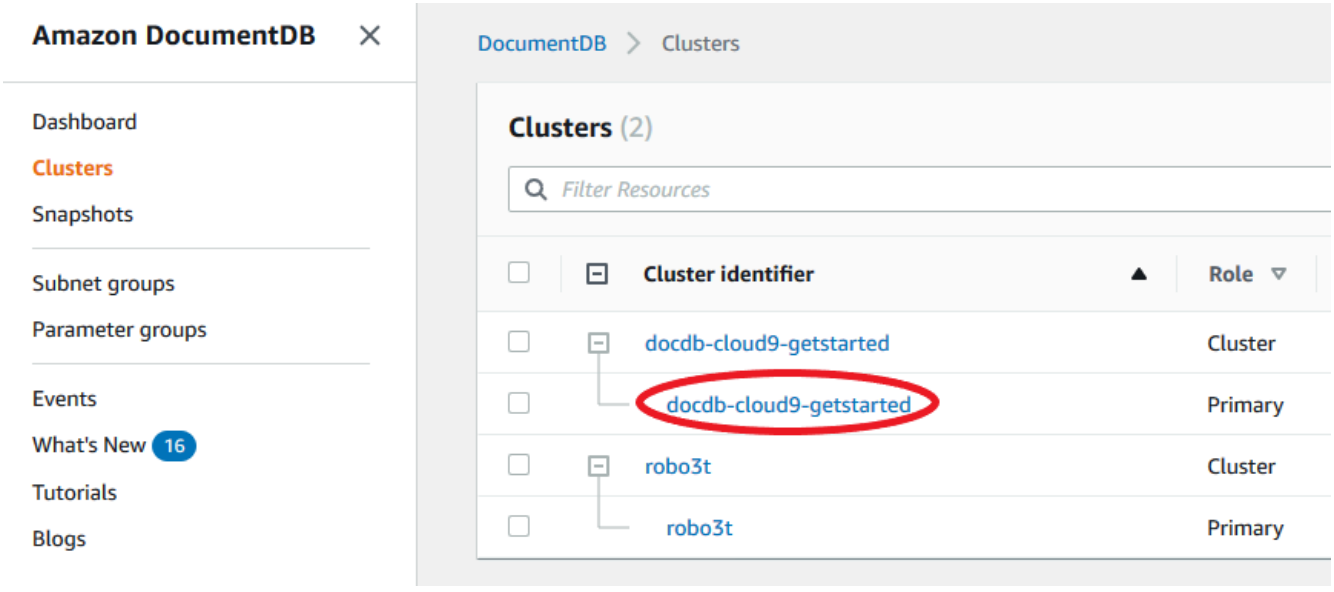
Diese CA-Zertifikate sind im regionalen und globalen Zertifikat-Bundle enthalten. Wenn Sie die rds-ca-rsa 2048-g1- oder rds-ca-rsa 4096-g1-CA mit einer Datenbank verwenden, verwaltet Amazon DocumentDB das DB-Serverzertifikat in der Datenbank. Amazon DocumentDB rotiert das DB-Serverzertifikat automatisch, bevor es abläuft (möglicherweise ist ein Neustart erforderlich).

> **Note**
>
> Das Aktualisieren Ihrer Instances erfordert einen Neustart, der zu Unterbrechungen des Service führen kann. Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

Using the AWS Management Console

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. Möglicherweise müssen Sie herausfinden, welche Instanzen sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2019`). Sie können dies in der Spalte Zertifizierungsstelle tun, die sich ganz rechts in der Cluster-Tabelle befindet.
5. In der Cluster-Tabelle sehen Sie ganz links die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



6. Markieren Sie das Kästchen links neben der Instanz, an der Sie interessiert sind.

7. Wählen Sie Aktionen und dann Ändern.
8. Wählen Sie unter Certificate authority (Zertifizierungsstelle) das neue Serverzertifikat (d. h. `rds-ca-rsa2048-g1`) für diese Instance aus.
9. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.
10. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden. Wenn Sie beabsichtigen, das Serverzertifikat sofort zu ändern, verwenden Sie die Option Apply Immediately (Sofort anwenden) .
11. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Using the AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS CLI

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Fehlerbehebung

Wenn im Rahmen der Zertifikatrotation Probleme beim Herstellen einer Verbindung mit dem Cluster auftreten, empfehlen wir Folgendes:

- Starten Sie Ihre Instances neu. Das Rotieren des neuen Zertifikats erfordert, dass Sie jede Ihrer Instances neu starten. Wenn Sie das neue Zertifikat auf eine oder mehrere Instances

angewendet haben, diese jedoch nicht neu gestartet haben, starten Sie die Instances neu, um das neue Zertifikat anzuwenden. Weitere Informationen finden Sie unter Neustarten einer Amazon DocumentDB-Instance.

- Stellen Sie sicher, dass Ihre Clients das neueste Zertifikatpaket verwenden. Siehe Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?.
- Stellen Sie sicher, dass Ihre Instances das neueste Zertifikat verwenden. Siehe Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?.
- Stellen Sie sicher, dass die neueste Zertifizierungsstelle von Ihrer Anwendung verwendet wird. Einige Treiber, wie Java und Go, benötigen zusätzlichen Code, um mehrere Zertifikate aus einem Zertifikatpaket in den Vertrauensspeicher zu importieren. Weitere Informationen zur Verbindung mit Amazon DocumentDB mit TLS finden Sie unterProgrammgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB.
- Wenden Sie sich an den Support. Wenn Sie Fragen oder Probleme haben, wenden Sie sich an AWS Support.

Häufig gestellte Fragen

Im Folgenden finden Sie Antworten auf einige häufig gestellte Fragen zu TLS-Zertifikaten.

Was passiert, wenn ich Fragen oder Probleme habe?

Wenn Sie Fragen oder Probleme haben, wenden Sie sich an AWS Support.

Woher weiß ich, ob ich TLS verwende, um eine Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Sie können bestimmen, ob Ihr Cluster TLS verwendet, indem Sie den `tls`-Parameter für die Clusterparametergruppe Ihres Clusters untersuchen. Wenn der `tls`-Parameter auf `enabled` festgelegt ist, verwenden Sie das TLS-Zertifikat, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

Warum aktualisieren Sie die Zertifizierungsstellen- und Serverzertifikate?

Die Amazon DocumentDB-CA- und Serverzertifikate werden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Die aktuellen CA- und Serverzertifikate laufen ab August 2024 ab.

Was passiert, wenn ich bis zum Ablaufdatum keine Maßnahmen ergreife?

Wenn Sie TLS für die Verbindung zu Ihrem Amazon DocumentDB-Cluster verwenden und die Zertifikatsänderung nicht bis zum vornehmen, können Ihre Anwendungen, die eine Verbindung über TLS herstellen, nicht mehr mit dem Amazon DocumentDB-Cluster kommunizieren.

Amazon DocumentDB rotiert Ihre Datenbankzertifikate vor Ablauf nicht automatisch. Sie müssen Ihre Anwendungen und Cluster aktualisieren, damit sie die neuen CA-Zertifikate vor oder nach dem Ablaufdatum verwenden können.

Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?

Um die Amazon DocumentDB-Instances zu identifizieren, die noch das alte Serverzertifikat verwenden, können Sie entweder die Amazon DocumentDB AWS Management Console oder die verwenden. AWS CLI

Verwenden der AWS Management Console

So identifizieren Sie die Instances in Ihren Clustern, die das ältere Zertifikat verwenden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Instances befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. In der Spalte Zertifizierungsstelle (ganz rechts in der Tabelle) wird angezeigt, welche Instanzen sich noch auf dem alten Serverzertifikat (`rds-ca-2019`) und dem neuen Serverzertifikat (`rds-ca-rsa2048-g1`) befinden.

Verwenden der AWS CLI

Um die Instances in Ihren Clustern zu identifizieren, die das ältere Serverzertifikat verwenden, verwenden Sie den Befehl `describe-db-clusters` mit Folgendem:

```
aws docdb describe-db-instances \
    --filters Name=engine,Values=docdb \
```

```
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Wie ändere ich einzelne Instances in meinem Amazon DocumentDB-Cluster, um das Serverzertifikat zu aktualisieren?

Sie sollten die Serverzertifikate für alle Instances in einem Cluster gleichzeitig aktualisieren. Um die Instances im Cluster zu ändern, können Sie die Konsole oder die AWS CLI verwenden.

> **Note**
>
> Das Aktualisieren Ihrer Instances erfordert einen Neustart, der zu Unterbrechungen des Service führen kann. Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

Verwenden der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. In der Spalte Zertifizierungsstelle (ganz rechts in der Tabelle) wird angezeigt, welche Instanzen sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2019`).
5. Wählen Sie in der Cluster-Tabelle unter Cluster-ID eine Instanz aus, die geändert werden soll.
6. Wählen Sie Aktionen und dann Ändern.
7. Wählen Sie unter Certificate authority (Zertifizierungsstelle) das neue Serverzertifikat (d. h. `rds-ca-rsa2048-g1`) für diese Instance aus.
8. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.
9. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden.
10. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Verwenden der AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS CLI

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Was passiert, wenn ich einem vorhandenen Cluster eine neue Instance hinzufüge?

Alle neu erstellten Instances verwenden das alte Serverzertifikat und erfordern TLS-Verbindungen über das alte CA-Zertifikat. Alle neuen Amazon DocumentDB DocumentDB-Instances, die nach dem 25. Januar 2024 erstellt wurden, verwenden standardmäßig das neue Zertifikat rds-ca-rsa 2048-g1.

Was passiert, wenn ein Instance-Ersatz oder ein Failover auf meinem Cluster vorhanden ist?

Wenn ein Instance-Ersatz in Ihrem Cluster vorhanden ist, verwendet die neu erstellte Instance weiterhin dasselbe Serverzertifikat wie die Instance davor. Es wird empfohlen, Serverzertifikate für alle Instances gleichzeitig zu aktualisieren. Wenn ein Failover im Cluster auftritt, wird das Serverzertifikat auf dem neuen Primärserver verwendet.

Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?

Wenn Sie TLS nicht verwenden, um eine Verbindung zu Ihren Amazon DocumentDB-Clustern herzustellen, sind keine Maßnahmen erforderlich.

Was soll ich tun, wenn ich derzeit nicht TLS für die Verbindung mit meinem Cluster verwende, dies zukünftig aber vorhabe?

Wenn Sie vor Januar 2024 einen Cluster erstellt haben, folgen Sie Schritt 1 und Schritt 2 im vorherigen Abschnitt, um sicherzustellen, dass Ihre Anwendung das aktualisierte CA-Bundle verwendet und dass jede Amazon DocumentDB DocumentDB-Instance das neueste Serverzertifikat verwendet. Wenn Sie nach dem 25. Januar 2024 einen Cluster erstellen, verfügt Ihr Cluster bereits über das neueste Serverzertifikat (rds-ca-rsa2048-g1). Informationen zum Überprüfen, ob Ihre Anwendung das neueste CA-Paket verwendet, finden Sie unter Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?.

Kann die Frist über den August 2024 hinaus verlängert werden?

Wenn Ihre Anwendungen eine Verbindung über TLS herstellen, kann die Frist nicht verlängert werden.

Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?

Verwenden Sie den folgenden Befehl, um zu überprüfen, ob Sie das neueste Paket haben. Um diesen Befehl ausführen zu können, muss Java installiert sein und die Java-Tools müssen sich in der PATH-Variablen Ihrer Shell befinden. Weitere Informationen finden Sie unter Java verwenden

macOS und Amazon Linux

```
keytool -printcert -v -file global-bundle.pem
```

Windows

```
keytool -printcert -v -file global-bundle.p7b
```

Warum sehe ich „RDS“ im Namen des Zertifizierungsstellenpakets?

Für bestimmte Verwaltungsfunktionen, wie z. B. die Zertifikatsverwaltung, verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.

Wann läuft das neue Zertifikat ab?

Das neue Serverzertifikat läuft (in der Regel) wie folgt ab:

- rds-ca-rsa2048-g1 — Läuft 2061 ab
- rds-ca-rsa4096-g1 — Läuft 2121 ab

Kann ich nach der Anwendung des neuen Serverzertifikats wieder zum alten Zertifikat zurückkehren?

Wenn Sie eine Instance auf das alte Serverzertifikat zurücksetzen müssen, sollten Sie alle Instances im Cluster zurücksetzen. Sie können das Serverzertifikat für jede Instance in einem Cluster mithilfe der AWS Management Console oder der AWS CLI zurücksetzen.

Verwenden der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. Wählen Sie in der Cluster-Tabelle unter Cluster-ID eine Instanz aus, die Sie ändern möchten. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
5. Unter Certificate authority (Zertifizierungsstelle) können Sie das alte Serverzertifikat (`rds-ca-2019`) auswählen.
6. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.
7. Auf dieser resultierenden Seite können Sie festlegen, dass Ihre Änderungen im nächsten Wartungsfenster oder sofort angewendet werden sollen. Wählen Sie die gewünschte Option und anschließend Modify instance (Instance ändern) aus.

> **Note**
>
> Wenn Sie Ihre Änderungen sofort anwenden möchten, werden alle Änderungen in der Warteschlange für ausstehende Änderungen ebenfalls angewendet. Wenn eine der ausstehenden Änderungen eine Ausfallszeit erfordert, kann die Auswahl dieser Option einen unerwarteten Ausfall verursachen.

Verwenden der AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2019 <--apply-immediately | --no-apply-immediately>
```

Wenn Sie `--no-apply-immediately` wählen, werden die Änderungen während des nächsten Wartungsfensters des Clusters angewendet.

Wird bei einer Wiederherstellung, die ich aus einem Snapshot oder zeitpunktbezogen ausführe, das neue Serverzertifikat verwendet?

Wenn Sie nach August 2024 einen Snapshot point-in-time wiederherstellen oder eine Wiederherstellung durchführen, verwendet der neue Cluster, der erstellt wird, das neue CA-Zertifikat.

Was ist, wenn ich Probleme habe, von einem beliebigen Mac OS aus eine direkte Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Mac OS hat die Anforderungen für vertrauenswürdige Zertifikate aktualisiert. Vertrauenswürdige Zertifikate müssen jetzt 397 Tage oder weniger gültig sein (siehehttps://support.apple.com/en-us/HT211025).

> **Note**
>
> Diese Einschränkung wird in neueren Versionen von Mac OS beachtet.

Amazon DocumentDB DocumentDB-Instance-Zertifikate sind über vier Jahre gültig und damit länger als das Mac OS-Maximum. Um von einem Computer mit Mac OS aus eine direkte Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, müssen Sie beim Erstellen der TLS-Verbindung ungültige Zertifikate zulassen. In diesem Fall bedeuten ungültige Zertifikate, dass die Gültigkeitsdauer mehr als 397 Tage beträgt. Sie sollten die Risiken verstehen, bevor Sie ungültige Zertifikate zulassen, wenn Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Verwenden Sie den Parameter, um von Mac OS aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellenAWS CLI, indem Sie den `tlsAllowInvalidCertificates` Parameter verwenden.

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate — GovCloud (US-West)

> **Note**
>
> Diese Information gilt nur für Benutzer in der Region GovCloud (US-West).

Das Zertifikat der Zertifizierungsstelle (CA) für Amazon DocumentDB-Cluster (mit MongoDB-Kompatibilität) wird am 18. Mai 2022 aktualisiert. Wenn Sie Amazon DocumentDB-Cluster mit aktiviertem Transport Layer Security (TLS) verwenden (Standardeinstellung) und Ihre Client-Anwendungs- und Serverzertifikate nicht rotiert haben, sind die folgenden Schritte erforderlich, um Verbindungsprobleme zwischen Ihrer Anwendung und Ihren Amazon DocumentDB-Clustern zu minimieren.

- Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung
- Schritt 2: Aktualisieren des Serverzertifikats

Die CA- und Serverzertifikate wurden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Das vorherige CA-Zertifikat läuft am 18. Mai 2022 ab. Client-Anwendungen müssen die neuen CA-Zertifikate zu ihren Trust Stores hinzufügen, und bestehende Amazon DocumentDB DocumentDB-Instances müssen aktualisiert werden, um die neuen CA-Zertifikate vor diesem Ablaufdatum zu verwenden.

Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters

Führen Sie die Schritte in diesem Abschnitt aus, um das CA-Zertifikatspaket Ihrer Anwendung (Schritt 1) und die Serverzertifikate Ihres Clusters (Schritt 2) zu aktualisieren. Bevor Sie die Änderungen auf Ihre Produktionsumgebungen anwenden, empfehlen wir dringend, diese Schritte in einer Entwicklungs- oder Stagingumgebung zu testen.

> **Note**
>
> Sie müssen die Schritte 1 und 2 jeweils ausführen,AWS-Region in denen Sie Amazon DocumentDB-Cluster haben.

Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung

Laden Sie das neue CA-Zertifikat herunter und aktualisieren Sie Ihre Anwendung, um das neue CA-Zertifikat zu verwenden, um TLS-Verbindungen zu Amazon DocumentDB herzustellen. Laden Sie das neue CA-Zertifikatpaket von https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem herunter. Diese Operation lädt die Datei mit dem Namen `us-gov-west-1-bundle.pem` herunter.

> **Note**
>
> Wenn Sie auf den Schlüsselspeicher zugreifen, der sowohl das alte CA-Zertifikat (`rds-ca-2017-root.pem`) als auch das neue CA-Zertifikat (`rds-ca-rsa4096-g1.pem`) enthält, müssen Sie sicherstellen, dass der Schlüsselspeicher `CA-RSA4096-G1` auswählt.

```
wget https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem
```

Aktualisieren Sie als Nächstes Ihre Anwendungen, um das neue Zertifikatpaket zu verwenden. Das neue CA-Bundle enthält sowohl das alte CA-Zertifikat als auch das neue CA-Zertifikat (`rds-ca-rsa4096-g1.pem`). Wenn beide CA-Zertifikate im neuen CA-Paket enthalten sind, können Sie Ihre Anwendung und Ihren Cluster in zwei Schritten aktualisieren.

Für alle Downloads des CA-Zertifikatspakets nach dem 21. Dezember 2021 sollte das neue CA-Zertifikatspaket verwendet werden. Informationen dazu, wie Sie überprüfen, ob Ihre Anwendung das neueste CA-Zertifikatspaket verwendet, finden Sie unter Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?. Wenn Sie das neueste CA-Zertifikatspaket in Ihrer Anwendung bereits verwenden, können Sie mit Schritt 2 fortfahren.

Beispiele für die Verwendung eines Zertifizierungsstellenpakets mit Ihrer Anwendung finden Sie unter Datenverschlüsselung während der Übertragung und Verbindung bei aktiviertem TLS herstellen.

> **Note**
>
> Zurzeit akzeptiert der MongoDB Go Driver 1.2.1 nur ein einzelnes CA-Serverzertifikat in `sslcertificateauthorityfile`. Informationen dazu, wie Sie bei aktiviertem TLS eine Verbindung zu Amazon DocumentDB über Go herstellen, finden Sie unter Verbindung bei aktiviertem TLS herstellen.

Schritt 2: Aktualisieren des Serverzertifikats

Nachdem die Anwendung aktualisiert wurde, um das neue CA-Paket zu verwenden, besteht der nächste Schritt darin, das Serverzertifikat zu aktualisieren, indem jede Instance in einem Amazon DocumentDB-Cluster geändert wird. Informationen zum Ändern von Instances zur Verwendung des neuen Serverzertifikats finden Sie in den folgenden Anweisungen.

> **Note**
>
> Das Aktualisieren Ihrer Instances erfordert einen Neustart, der zu Unterbrechungen des Service führen kann. Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

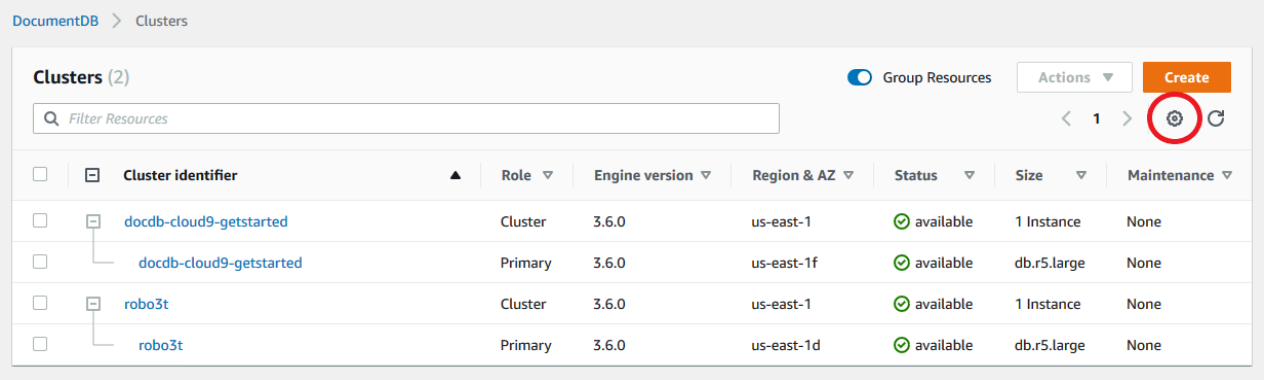
Using the AWS Management Console

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances zu identifizieren und zu rotieren. Verwenden Sie dazu denAWS Management Console.

1. Melden Sie sich bei der dieAWS Management Console Amazon DocumentDB-Konsole an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der die die -die -die -die -die -die -die -die -die -die -die -die -die -dieAWS-Region -die -die -die -die die -die die -die die
3. wh

   Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf der linken Seite der Konsole auf der die -die die -die die -die die -Konsole
4. Möglicherweise müssen Sie herausfinden, welche Instanzen sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2017`). Sie können dies in der Spalte Zertifizierungsstelle tun, die standardmäßig ausgeblendet ist. Gehen Sie folgendermaßen vor, um die Spalte Certificate authority (Zertifizierungsstelle) anzuzeigen:

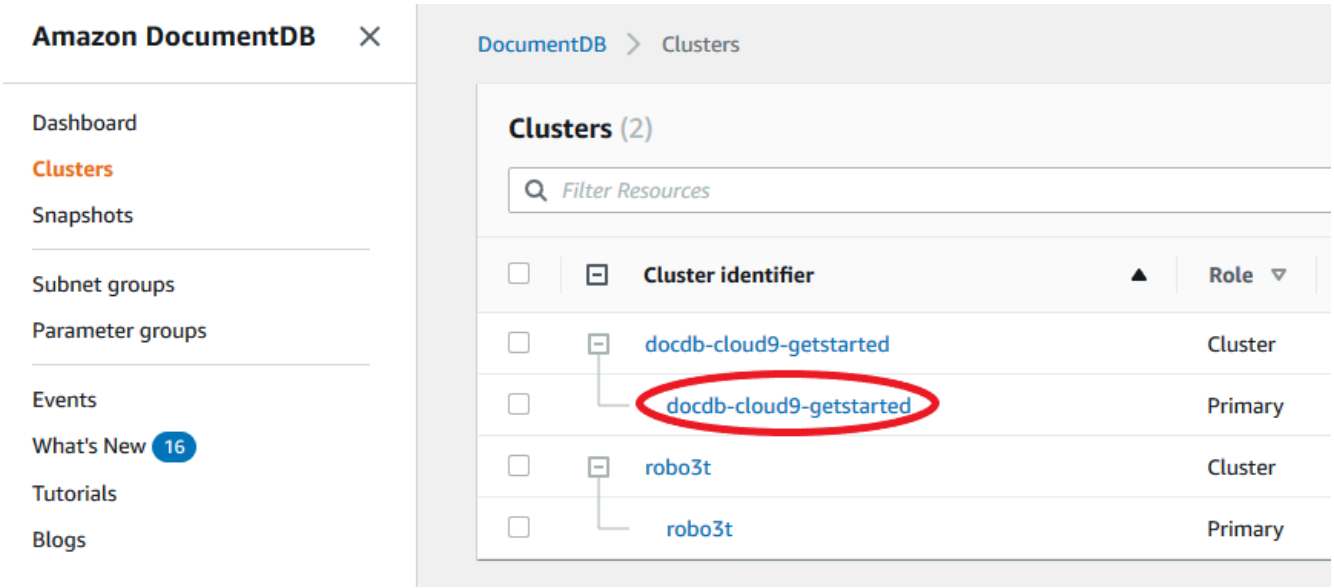
   a. Wählen Sie das Symbol Settings (Einstellungen).



b. Wählen Sie in der Liste der sichtbaren Spalten die Spalte Certificate authority (Zertifizierungsstelle).

c. Wählen Sie Confirm (Bestätigen), um Ihre Änderungen zu speichern.

5. Zurück im Cluster-Navigationsfeld sehen Sie nun die Spalte Cluster Identifier. Ihre Instances sind unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



6. Aktivieren Sie das Kontrollkästchen links neben der die -Instance, die Sie interessiert.

7. Wählen Sie Aktionen und dann Ändern.

8. Wählen Sie unter Certificate authority (Zertifizierungsstelle) das neue Serverzertifikat (d. h. `rds-ca-rsa4096-g1`) für diese Instance aus.

9. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.

10. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden. Wenn Sie beabsichtigen, das Serverzertifikat sofort zu ändern, verwenden Sie die Option Apply Immediately (Sofort anwenden) .
11. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Using the AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances zu identifizieren und zu rotieren. Verwenden Sie dazu denAWS CLI.

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Fehlerbehebung

Wenn im Rahmen der Zertifikatrotation Probleme beim Herstellen einer Verbindung mit dem Cluster auftreten, empfehlen wir Folgendes:

- Starten Sie Ihre Instances neu. Das Rotieren des neuen Zertifikats erfordert, dass Sie jede Ihrer Instances neu starten. Wenn Sie das neue Zertifikat auf eine oder mehrere Instances angewendet haben, diese jedoch nicht neu gestartet haben, starten Sie die Instances neu, um das neue Zertifikat anzuwenden. Weitere Informationen finden Sie unter Neustarten einer Amazon DocumentDB-Instance.
- Stellen Sie sicher, dass Ihre Clients das neueste Zertifikatpaket verwenden. Siehe Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?.

- Stellen Sie sicher, dass Ihre Instances das neueste Zertifikat verwenden. Siehe Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?.
- Stellen Sie sicher, dass die neueste Zertifizierungsstelle von Ihrer Anwendung verwendet wird. Einige Treiber, wie Java und Go, benötigen zusätzlichen Code, um mehrere Zertifikate aus einem Zertifikatpaket in den Vertrauensspeicher zu importieren. Weitere Informationen zum Herstellen einer Verbindung mit Amazon DocumentDB mit TLS finden Sie unterProgrammgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB.
- Wenden Sie sich an den Support. Wenn Sie Fragen oder Probleme haben, wenden Sie sich an AWS Support.

Häufig gestellte Fragen

Im Folgenden finden Sie Antworten auf einige häufig gestellte Fragen zu TLS-Zertifikaten.

Was passiert, wenn ich Fragen oder Probleme habe?

Wenn Sie Fragen oder Probleme haben, wenden Sie sich an AWS Support.

Woher weiß ich, ob ich TLS verwende, um eine Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Sie können bestimmen, ob Ihr Cluster TLS verwendet, indem Sie den `tls`-Parameter für die Clusterparametergruppe Ihres Clusters untersuchen. Wenn der `tls`-Parameter auf `enabled` festgelegt ist, verwenden Sie das TLS-Zertifikat, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

Warum aktualisieren Sie die Zertifizierungsstellen- und Serverzertifikate?

Die CA- und Serverzertifikate von Amazon DocumentDB wurden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Die aktuellen CA- und Serverzertifikate laufen am Mittwoch, den 18. Mai 2022 ab.

Was passiert, wenn ich bis zum Ablaufdatum durch?

Wenn Sie TLS verwenden, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, und Sie die Änderung nicht bis zum 18. Mai 2022 vornehmen, können Ihre Anwendungen, die sich über TLS verbinden, nicht mehr mit dem Amazon DocumentDB-Cluster kommunizieren.

Amazon DocumentDB rotiert Ihre Datenbankzertifikate vor Ablauf nicht automatisch. Sie müssen Ihre Anwendungen und Cluster aktualisieren, um die neuen CA-Zertifikate vor oder nach dem Ablaufdatum verwenden zu können.

Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?

Um die Amazon DocumentDB-Instances zu identifizieren, die noch das alte Serverzertifikat verwenden, können Sie entweder die Amazon DocumentDBAWS Management Console oder die verwendenAWS CLI.

Verwendung der AWS Management Console

So identifizieren Sie die Instances in Ihren Clustern, die das ältere Zertifikat verwenden

1. Melden Sie sich bei der dieAWS Management Console Amazon DocumentDB-Konsole an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der die die -die -die -die -die -die -die -die -die -die -die die -Instance aus,AWS-Region in der die die die -Instance aus.
3. Klicken Sie im Navigationsbereich links in der Konsole auf Instances (Instances).
4. Die Spalte Certificate authority (Zertifizierungsstelle) (standardmäßig ausgeblendet) zeigt an, welche Instances sich noch auf dem alten Serverzertifikat (`rds-ca-2017`) und auf dem neuen Serverzertifikat (`rds-ca-rsa4096-g1`) befinden. Gehen Sie folgendermaßen vor, um die Spalte Certificate authority (Zertifizierungsstelle) anzuzeigen:
   a. Wählen Sie das Symbol Settings (Einstellungen).
   b. Wählen Sie in der Liste der sichtbaren Spalten die Spalte Certificate authority (Zertifizierungsstelle).
   c. Wählen Sie Confirm (Bestätigen), um Ihre Änderungen zu speichern.

Verwendung der AWS CLI

Um die Instances in Ihren Clustern zu identifizieren, die das ältere Serverzertifikat verwenden, verwenden Sie den Befehl `describe-db-clusters` mit Folgendem:

```
aws docdb describe-db-instances \
```

```
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Wie ändere ich einzelne Instances in meinem Amazon DocumentDB-Cluster, um das Serverzertifikat zu aktualisieren?

Sie sollten die Serverzertifikate für alle Instances in einem Cluster gleichzeitig aktualisieren. Um die Instances im Cluster zu ändern, können Sie die Konsole oder die AWS CLI verwenden.

> **Note**
>
> Das Aktualisieren Ihrer Instances erfordert einen Neustart, der zu Unterbrechungen des Service führen kann. Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

Verwendung der AWS Management Console

1. Melden Sie sich bei der dieAWS Management Console Amazon DocumentDB-Konsole an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der die die -die -die -die -die -die -die -die -die -die -die -die -die -dieAWS-Region -die -die -die -die die -die die -die die
3. Klicken Sie im Navigationsbereich links in der Konsole auf Instances (Instances).
4. Die Spalte Certificate authority (Zertifizierungsstelle) (standardmäßig ausgeblendet) zeigt an, welche Instances sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2017`). Gehen Sie folgendermaßen vor, um die Spalte Certificate authority (Zertifizierungsstelle) anzuzeigen:
   a. Wählen Sie das Symbol Settings (Einstellungen).
   b. Wählen Sie in der Liste der sichtbaren Spalten die Spalte Certificate authority (Zertifizierungsstelle).
   c. Wählen Sie Confirm (Bestätigen), um Ihre Änderungen zu speichern.
5. Wählen Sie eine zu ändernde Instance aus.
6. Wählen Sie Aktionen und dann Ändern.

7. Wählen Sie unter Zertifizierungsstelle das neue Serverzertifikat (`rds-ca-rsa4096-g1`) für diese Instanz aus.
8. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.
9. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden.
10. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Verwendung der AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances zu identifizieren und zu rotieren. Verwenden Sie dazu denAWS CLI.

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Was passiert, wenn ich einem vorhandenen Cluster eine neue Instance hinzufüge?

Alle neu erstellten Instances verwenden das alte Serverzertifikat und erfordern TLS-Verbindungen über das alte CA-Zertifikat. Alle neuen Amazon DocumentDB DocumentDB-Instances, die nach dem 21. März 2022 erstellt wurden, verwenden standardmäßig die neuen Zertifikate.

Was passiert, wenn ein Instance-Ersatz oder ein Failover auf meinem Cluster vorhanden ist?

Wenn ein Instance-Ersatz in Ihrem Cluster vorhanden ist, verwendet die neu erstellte Instance weiterhin dasselbe Serverzertifikat wie die Instance davor. Es wird empfohlen, Serverzertifikate für alle Instances gleichzeitig zu aktualisieren. Wenn ein Failover im Cluster auftritt, wird das Serverzertifikat auf dem neuen Primärserver verwendet.

Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?

Wenn Sie TLS nicht verwenden, um eine Verbindung zu Ihren Amazon DocumentDB-Clustern herzustellen, sind keine Maßnahmen erforderlich.

Was soll ich tun, wenn ich derzeit nicht TLS für die Verbindung mit meinem Cluster verwende, dies zukünftig aber vorhabe?

Wenn Sie vor dem 21. März 2022 einen Cluster erstellt haben, folgen Sie Schritt 1 und Schritt 2 im vorherigen Abschnitt, um sicherzustellen, dass Ihre Anwendung das aktualisierte CA-Bundle verwendet und dass jede Amazon DocumentDB DocumentDB-Instance das neueste Serverzertifikat verwendet. Wenn Sie nach dem 21. März 2022 einen Cluster erstellen, verfügt Ihr Cluster bereits über das neueste Serverzertifikat. Informationen zum Überprüfen, ob Ihre Anwendung das neueste CA-Paket verwendet, finden Sie unter Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?.

Kann die Frist über den 18. Mai 2022 hinaus verlängert werden?

Wenn Ihre Bewerbungen über TLS verbunden werden, kann die Frist nicht über den 18. Mai 2022 hinaus verlängert werden.

Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?

Aus Kompatibilitätsgründen werden sowohl alte als auch neue Zertifizierungsstellenpaket-Dateien mit `us-gov-west-1-bundle.pem` benannt. Darüber hinaus können Sie Tools wie `openssl` oder `keytool` verwenden, um das CA-Paket zu überprüfen.

Warum sehe ich „RDS“ im Namen des Zertifizierungsstellenpakets?

Für bestimmte --Verwaltungsfunktionen, wie die -Zertifikatsverwaltung, verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.

Kann ich nach der Anwendung des neuen Serverzertifikats wieder zum alten Zertifikat zurückkehren?

Wenn Sie eine Instance auf das alte Serverzertifikat zurücksetzen müssen, sollten Sie alle Instances im Cluster zurücksetzen. Sie können das Serverzertifikat für jede Instance in einem Cluster mithilfe der AWS Management Console oder der AWS CLI zurücksetzen.

Verwendung der AWS Management Console

1. Melden Sie sich bei der dieAWS Management Console Amazon DocumentDB-Konsole an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der die die -die -die -die -die -die -die -die -die -die -die -die -die -dieAWS-Region -die -die -die -die die -die die -die die
3. Klicken Sie im Navigationsbereich links in der Konsole auf Instances (Instances).
4. Wählen Sie eine zu ändernde Instance aus. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
5. Unter Zertifizierungsstelle können Sie das alte Serverzertifikat (`rds-ca-2017`) auswählen.
6. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.
7. Auf dieser resultierenden Seite können Sie festlegen, dass Ihre Änderungen im nächsten Wartungsfenster oder sofort angewendet werden sollen. Wählen Sie die gewünschte Option und anschließend Modify instance (Instance ändern) aus.

> **Note**
>
> Wenn Sie Ihre Änderungen sofort anwenden möchten, werden alle Änderungen in der Warteschlange für ausstehende Änderungen ebenfalls angewendet. Wenn eine der ausstehenden Änderungen eine Ausfallszeit erfordert, kann die Auswahl dieser Option einen unerwarteten Ausfall verursachen.

Verwendung der AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2017 <--apply-immediately | --no-apply-immediately>
```

Wenn Sie `--no-apply-immediately` wählen, werden die Änderungen während des nächsten Wartungsfensters des Clusters angewendet.

Wird bei einer Wiederherstellung, die ich aus einem Snapshot oder zeitpunktbezogen ausführe, das neue Serverzertifikat verwendet?

Wenn Sie nach dem 21. März 2022 einen Snapshot point-in-time wiederherstellen oder eine Wiederherstellung durchführen, verwendet der neu erstellte Cluster das neue CA-Zertifikat.

Was ist, wenn ich Probleme habe, von Mac OS X Catalina aus eine direkte Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Mac OS X Catalina hat die Anforderungen für vertrauenswürdige Zertifikate aktualisiert. Vertrauenswürdige Zertifikate müssen jetzt 825 Tage oder weniger gültig sein (siehehttps://support.apple.com/en-us/HT210176). Amazon DocumentDB DocumentDB-Instanzzertifikate sind über vier Jahre gültig und damit länger als das Maximum für Mac OS X. Um von einem Computer aus, auf dem Mac OS X Catalina ausgeführt wird, eine direkte Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, müssen Sie beim Erstellen der TLS-Verbindung ungültige Zertifikate zulassen. In diesem Fall bedeuten ungültige Zertifikate, dass die Gültigkeitsdauer länger als 825 Tage beträgt. Sie sollten die Risiken verstehen, bevor Sie ungültige Zertifikate zulassen, wenn Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Verwenden Sie den`tlsAllowInvalidCertificates` Parameter, um von OS X Catalina aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen.AWS CLI

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Konformitätsprüfung in Amazon DocumentDB

Die Sicherheit und Compliance von Amazon DocumentDB (mit MongoDB-Kompatibilität) wird von externen Prüfern im Rahmen mehrererAWS -Compliance-Programme bewertet. Dazu gehören auch folgende:

- System and Organization Controls (SOC) 1, 2 und 3. Weitere Informationen finden Sie unter SOC.
- Payment Card Industry Data Security Standard (PCI DSS). Weitere Informationen finden Sie unter PCI DSS.
- ISO 9001, 27001, 27017 und 27018. Weitere Informationen finden Sie unter ISO-zertifiziert.
- Health Insurance Portability and Accountability Act Business Associate Agreement (HIPAA BAA). Weitere Informationen finden Sie unter HIPAA-Compliance.

AWS bietet eine häufig aktualisierte Liste der AWS-Services, die von bestimmten Compliance-Programmen verwendet werden, unter AWS-Services im Bereich von Compliance-Programmen.

Audit-Berichte von Drittanbietern stehen Ihnen zum Download über AWS Artifact zur Verfügung. Weitere Informationen finden Sie unter Herunterladen von Berichten in AWS-Artifact.

Weitere Informationen zu AWS-Compliance-Programmen finden Sie unter AWS-Compliance-Programme.

Welche Compliance-Verpflichtungen Sie bei der Verwendung von Amazon DocumentDB haben, hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihrer Organisation und den geltenden Gesetzen und Vorschriften ab. Wenn Ihre Nutzung von Amazon DocumentDB der Einhaltung von Standards wie HIPAA oder PCI unterliegt,AWS stellt Ressourcen zur Unterstützung bereit:

- AWS-Compliance-Ressourcen – Eine Sammlung von Arbeitsmappen und Leitfäden, die für Ihre Branche und Ihren Standort interessant sein könnte.
- Schnellstartleitfaden für Sicherheit und Compliance – Bereitstellungsanleitungen, die Überlegungen zur Architektur enthalten und Schritte für die Bereitstellung von Basisumgebungen mit dem Schwerpunkt auf Sicherheit und Compliance in AWS beschreiben.
- AWSConfig: Ein Service, der die Konformität Ihrer Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften bewertet.
- AWS Security Hub – Eine umfassende Ansicht Ihres Sicherheitsstatus innerhalb von AWS, die Ihnen hilft, Ihre Compliance mit Standards und bewährten Methoden der Sicherheitsbranche zu überprüfen.
- Whitepaper zum Thema Architekturen für Sicherheit und Compliance im Zusammenhang mit HIPAA: Ein Whitepaper, in dem beschrieben wird, wie Unternehmen verwenden können,AWS um HIPAA-konforme Anwendungen zu erstellen.

Ausfallsicherheit in Amazon DocumentDB

Die globale AWS-Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit geringer Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Ein Amazon DocumentDB DocumentDB-Cluster kann nur in einer Amazon VPC erstellt werden, die mindestens über zwei Subnetze in mindestens zwei Availability Zones verfügt. Durch die Verteilung Ihrer Cluster-Instances über mindestens zwei Availability Zones trägt Amazon DocumentDB dazu bei, sicherzustellen, dass im DB-Cluster auch im unwahrscheinlichen Fall eines Availability Zone-Ausfalls Instances verfügbar bleiben. Das Cluster-Volume für den Amazon DocumentDB DocumentDB-Cluster wird stets auf drei Availability Zones ausgedehnbar, um belastbaren Speicher mit einem geringeren Datenverlustr

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter AWSGlobale Infrastruktur.

Darüber hinaus gilt:AWSAmazon DocumentDB stellt verschiedene Funktionen bereit, um Ihren Anforderungen an Ausfallsicherheit und Datensicherung gerecht zu werden.

Fehlertolerante Speicherung, die Probleme automatisch behebt

> Jeder 10-GB-Teil Ihres Speicher-Volumes wird auf sechs Arten über drei Availability Zones repliziert. Amazon DocumentDB verwendet fehlertoleranten Speicher, der den Verlust von bis zu zwei Datenkopien transparent behandelt, ohne die Verfügbarkeit von Datenbanken zu beeinträchtigen, und bis zu drei Kopien, ohne die Leseverfügbarkeit zu beeinträchtigen. Amazon DocumentDB DocumentDB-Speicher ist ebenfalls selbstheilend; Datenblöcke und Festplatten werden kontinuierlich auf Fehler gescannt und automatisch ausgetauscht.

Manuelle Sicherungen und Wiederherstellung

> Amazon DocumentDB bietet die Möglichkeit, vollständige Sicherungen Ihres Clusters für die langfristige Aufbewahrung und Wiederherstellung zu erstellen. Weitere Informationen finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.

Zeitpunktbezogene Wiederherstellung

Mit der zeitpunktbezogenen Wiederherstellung schützen Sie Ihre Amazon DocumentDB DocumentDB-Cluster vor versehentlichen Schreib- und Löschoperationen. Mit der zeitpunktbezogenen Wiederherstellung müssen Sie sich keine Gedanken über das Erstellen, Warten oder Planen von On-Demand-Backups machen. Weitere Informationen finden Sie unter Wiederherstellen auf einen bestimmten Zeitpunkt.

Infrastruktursicherheit in Amazon DocumentDB

Als verwalteter Service ist Amazon DocumentDB geschützt durchAWSglobale Netzwerksicherheit. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter AWSCloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastrukturschutz im Security Pillar AWS Well-Architected Framework.

Du verwendestAWSveröffentlichte API-Aufrufe für den Zugriff auf Amazon DocumentDB über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sie können diese API-Vorgänge von einem beliebigen Netzwerkstandort aus aufrufen. Sie können Amazon DocumentDB-Richtlinien verwenden, um den Zugriff von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten VPCs aus zu steuern. Dadurch wird der Netzwerkzugriff auf eine bestimmte Amazon DocumentDB-Ressource effektiv von nur der spezifischen VPC innerhalb derAWSnetzwerk.

> **Note**
>
> Amazon DocumentDB unterstützt keine ressourcenbasierten Zugriffsrichtlinien.

Bewährte Sicherheitsmethoden für Amazon DocumentDB

Für Best Practices für die Sicherheit müssen SieAWS Identity and Access Management(IAM) - Konten zur Steuerung des Zugriffs auf Amazon DocumentDB DocumentDB-API-Operationen, insbesondere Operationen, mit denen Amazon DocumentDB DocumentDB-Ressourcen erstellt, geändert oder gelöscht werden. Zu solchen Ressourcen gehören Cluster, Sicherheitsgruppen und Parametergruppen. Sie müssen auch IAM verwenden, um Aktionen zu steuern, mit denen allgemeine administrative Aktionen durchgeführt werden, z. B. das Sichern der Wiederherstellung von Clustern. Wenden Sie beim Erstellen von IAM-Rollen das Prinzip der geringsten Rechte an.

- Erzwingen Sie die geringste Berechtigung mit rollenbasierter Zugriffssteuerung.
- Weisen Sie jeder Person, die Amazon DocumentDB DocumentDB-Ressourcen verwaltet, ein eigenes IAM-Konto zu. Verwenden Sie nicht dasAWS-KontoRoot-Benutzer zur Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen. Erstellen Sie einen IAM-Benutzer für jede Person einschließlich Sie selbst.
- Gewähren Sie jedem -Benutzer nur den Mindestsatz an Berechtigungen, die für die Ausführung seiner Aufgaben erforderlich sind.
- Verwenden Sie IAM-Gruppen, um Berechtigungen für mehrere Benutzer effektiv zu verwalten. Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch. Weitere Informationen zu bewährten Methoden für IAM finden Sie unter Bewährte Methoden für IAM.
- Wechseln Sie regelmäßig die IAM-Anmeldeinformationen.
- KonfigurationAWSSecrets Manager zum automatischen Drehen der Secrets für Amazon DocumentDB. Weitere Informationen finden Sie unterDrehen vonAWSSecrets Manager-SecretsundRotieren von geheimen Schlüsseln für Amazon DocumentDBimAWSSecrets Manager Manager-Pataus.
- Verwenden Sie Transport Layer Security (TLS) und die Verschlüsselung ruhender Daten, um Ihre Daten zu verschlüsseln.

Amazon DocumentDB DocumentDB-Ereignisse prüfen

Mit Amazon DocumentDB (mit MongoDB-Kompatibilität) können Sie Ereignisse überprüfen, die in Ihrem Cluster durchgeführt wurden. Beispiele für protokollierte Ereignisse sind erfolgreiche und fehlgeschlagene Authentifizierungsversuche, Drop-Ereignisse für Sammlungen in einer Datenbank oder das Erstellen eines Index. Standardmäßig ist die Prüfung in Amazon DocumentDB deaktiviert und erfordert, dass Sie sich für die Nutzung dieser Funktion anmelden.

Wenn die Prüfung aktiviert ist, zeichnet Amazon DocumentDB Ereignisse in Data Definition Language (DDL), Data Manipulation Language (DML), Authentifizierung, Autorisierung und Benutzerverwaltung in Amazon Logs auf. CloudWatch Wenn Auditing aktiviert ist, exportiert Amazon DocumentDB die Auditing-Datensätze (JSON-Dokumente) Ihres Clusters nach Amazon CloudWatch Logs. Sie können Amazon CloudWatch Logs verwenden, um Ihre Amazon DocumentDB-Prüfungsereignisse zu analysieren, zu überwachen und zu archivieren.

Amazon DocumentDB berechnet zwar keine zusätzlichen Kosten für die Aktivierung der Prüfung, Ihnen werden jedoch Standardtarife für die Nutzung von CloudWatch Logs berechnet. Informationen zu den Preisen für CloudWatch Logs finden Sie unter CloudWatch Amazon-Preise.

Die Amazon DocumentDB-Prüfungsfunktion unterscheidet sich deutlich von der Nutzung der Serviceressourcen, mit der überwacht wird. AWS CloudTrail CloudTrail zeichnet Operationen AWS Management Console auf, die mit AWS Command Line Interface (AWS CLI) oder mit Ressourcen wie Clustern, Instances, Parametergruppen und Snapshots ausgeführt werden. Die Überwachung von AWS Ressourcen mit CloudTrail ist standardmäßig aktiviert und kann nicht deaktiviert werden. Die Amazon DocumentDB-Prüfungsfunktion ist eine optionale Funktion. Sie zeichnet Operationen auf, die innerhalb Ihres Clusters für Objekte, wie z. B. Datenbanken, Sammlungen, Indizes und Benutzer ausgeführt werden.

Themen

Unterstützte Ereignisse

Amazon DocumentDB DocumentDB-Auditing unterstützt die folgenden Ereigniskategorien:

- Data Definition Language (DDL) — umfasst Datenbankverwaltungsvorgänge, Verbindungen, Benutzerverwaltung und Autorisierung.
- Leseereignisse (DML-Lesevorgänge) in der Data Manipulation Language — umfasst die verschiedenen Aggregationsoperatoren, arithmetischen Operatoren, booleschen Operatoren `find()` und andere Leseabfrageoperatoren.
- Schreibereignisse (DML-Schreibvorgänge) in Data Manipulation Language — beinhaltet Operatoren und `insert(), update(), delete(), bulkWrite()`

Folgende Ereignistypen werden unterstützt:

Ereignistyp	Kategorie	Beschreibung
`authCheck`	Autorisierung	Ergebniscode 0: Erfolg
		Ergebniscode 13: Unbefugte Versuche, einen Vorgang auszuführen.
`authenticate`	Verbindung	Erfolgreiche oder fehlgeschlagene Authentifizierungs versuche bei einer neuen Verbindung.
`createDatabase`	DDL	Erstellung einer neuen Datenbank.
`createCollection`	DDL	Erstellung einer neuen Sammlung innerhalb einer Datenbank.
`createIndex`	DDL	Erstellung eines neuen Index innerhalb einer Sammlung.

Ereignistyp	Kategorie	Beschreibung
`dropCollection`	DDL	Löschen einer Sammlung in einer Datenbank.
`dropDatabase`	DDL	Löschen einer Datenbank.
`dropIndex`	DDL	Löschen eines Index innerhalb einer Sammlung.
`modifyChangeStreams`	DDL	Der Change-Stream wurde erstellt.
`renameCollection`	DDL	Umbenennen einer Sammlung innerhalb einer Datenbank.
`createRole`	Rollenverwaltung	Eine Rolle erstellen.
`dropAllRolesFromDatabase`	Rollenverwaltung	Alle Rollen innerhalb einer Datenbank löschen.
`dropRole`	Rollenverwaltung	Eine Rolle löschen.
`grantPrivilegesToRole`	Rollenverwaltung	Einer Rolle Privilegien gewähren
`grantRolesToRole`	Rollenverwaltung	Einer benutzerd efinierten Rolle Rollen zuweisen
`revokePrivilegesFromRole`	Rollenverwaltung	Rechte einer Rolle entziehen

Ereignistyp	Kategorie	Beschreibung
`revokeRolesFromRole`	Rollenverwaltung	Sperren von Rollen aus einer benutzerd efinierten Rolle
`updateRole`	Rollenverwaltung	Eine Rolle aktualisi eren.
`createUser`	Benutzerverwaltung	Anlegen eines neuen Benutzers.
`dropAllUsersFromDatabase`	Benutzerverwaltung	Löschen aller Benutzer innerhalb einer Datenbank.
`dropUser`	Benutzerverwaltung	Löschen eines bestehenden Benutzers.
`grantRolesToUser`	Benutzerverwaltung	Einem Benutzer Rollen zuweisen.
`revokeRolesFromUser`	Benutzerverwaltung	Einem Benutzer Rollen entziehen.
`updateUser`	UserManagement	Aktualisierung eines bestehenden Benutzers.
`insert`	DML schreiben	Fügt ein oder mehrere Dokumente in eine Sammlung ein.
`delete`	DML, schreiben	Löscht ein oder mehrere Dokumente aus einer Sammlung.

Ereignistyp	Kategorie	Beschreibung
`update`	DML, schreiben	Ändert ein vorhandenes Dokument oder Dokumente in einer Sammlung.
`bulkWrite`	DML-Schreiben	Führt mehrere Schreiboperationen mit Steuerelementen für die Reihenfolge der Ausführung aus.
`count`	DML gelesen	Gibt die Anzahl der Dokumente zurück, die einer find () -Abfrage für die Sammlung oder Ansicht entsprechen würden.
`countDocuments`	DML gelesen	Gibt die Anzahl der Dokumente zurück, die der Abfrage für eine Sammlung oder Ansicht entsprechen.
`find`	DML gelesen	Wählt Dokumente in einer Sammlung oder Ansicht aus und bringt einen Cursor zu den ausgewählten Dokumenten zurück.
`findAndModify`	DML lesen und DML schreiben	Ändert ein einzelnes Dokument und gibt es zurück.

Ereignistyp	Kategorie	Beschreibung
`findOneAndDelete`	DML lesen und DML schreiben	Löscht ein einzelnes Dokument auf der Grundlage der Filter- und Sortierkr iterien und gibt das gelöschte Dokument zurück.
`findOneAndReplace`	DML lesen und DML schreiben	Ersetzt ein einzelnes Dokument auf der Grundlage des angegebenen Filters.
`findOneAndUpdate`	DML lesen und DML schreiben	Aktualisiert ein einzelnes Dokument auf der Grundlage der Filter- und Sortierkr iterien.
`aggregate`	DML lesen und DML schreiben	Unterstützt APIs in der Aggregationspipeli ne.
`distinct`	DML gelesen	Findet die unterschi edlichen Werte für ein bestimmtes Feld in einer einzelnen Sammlung oder Ansicht und gibt die Ergebnisse in einem Array zurück.

> **Note**
>
> Die Werte im Parameterfeld des DML-Ereignisdokuments haben eine Größenbeschränkung von 1 KB. Amazon DocumentDB kürzt den Wert, wenn er 1 KB überschreitet.

> **Note**
>
> TTL-Löschereignisse werden derzeit nicht geprüft.

Aktivieren des Prüfens

Die Aktivierung des Prüfens für einen Cluster ist ein zweistufiger Prozess. Stellen Sie sicher, dass beide Schritte abgeschlossen sind, da andernfalls keine Auditprotokolle an Logs gesendet CloudWatch werden.

Schritt 1. Aktivieren Sie den Clusterparameter audit_logs

Um die Überwachung zu aktivieren, müssen Sie den `audit_logs` Parameter in der Parametergruppe ändern. `audit_logs`ist eine durch Kommas getrennte Liste von Ereignissen, die protokolliert werden sollen. Ereignisse müssen in Kleinbuchstaben angegeben werden und es darf kein Leerzeichen zwischen den Listenelementen sein.

Sie können die folgenden Werte für die Parametergruppe festlegen:

Wert	Beschreibung	
`ddl`	Diese Einstellung ermöglicht die Überwachung von DDL-Ereignissen wie CreateDatabase, DropDatabase, CreateCollection, DropCollection, CreateIndex, DropIndex,	

Wert	Beschreibung	
	AuthCheck, authentic ate, createUser, DropUser, User, updateUser und grantRolesTo revokeRolesFrom dropAllUsers FromDatabase	
`dml_read`	Diese Einstellu ng ermöglicht die Prüfung von DML-Leseereignissen wie Find, Sort Count, Distinct, Group, Projecta, Unwind, GeoNear, GeoInters ects, GeoWithin und anderen MongoDB-L eseabfrageoperator en.	
`dml_write`	Wenn Sie diese Einstellung festlegen , wird die Überwachu ng von DML-Schre ibereignissen wie insert (), update (), delete () und bulkWrite () aktiviert	

Wert	Beschreibung	
`all`	Wenn Sie diese Einstellung festlegen, wird die Überwachung Ihrer Datenbankereignisse wie Leseabfragen, Schreibabfragen, Datenbankaktionen und Administratoraktionen aktiviert.	
none	Wenn Sie diese Einstellung festlegen, wird die Überwachung deaktiviert	

Wert	Beschreibung	
`enabled` (veraltet)	Dies ist eine ältere Parametereinstellung, die 'ddl' entspricht. Diese Einstellung ermöglicht die Überwachung von DDL-Ereignissen wie CreateDatabase, DropDatabase, CreateCollection, DropCollection, CreateIndex, DropIndex, AuthCheck, authenticate, createUser, DropUser, User, updateUser und. grantRolesTo revokeRolesFrom dropAllUsersFromDatabase Es wird nicht empfohlen, diese Einstellung zu verwenden, da es sich um eine ältere Einstellung handelt.	

Wert	Beschreibung
`disabled` (veraltet)	Dies ist eine ältere Parametereinstellung, die „none“ entspricht. Wir empfehlen, diese Einstellung nicht zu verwenden, da es sich um eine ältere Einstellung handelt.

> **Note**
>
> Der Standardwert für den Clusterparameter audit_logs ist `none` (legacy "`disabled`„).

Sie können die oben genannten Werte auch in Kombinationen verwenden.

Wert	Beschreibung
`ddl, dml_read`	Wenn Sie diese Einstellung festlegen, wird die Überwachung von DDL-Ereignissen und DML-Lesee reignissen aktiviert.
`ddl, dml_write`	Diese Einstellu ng aktiviert die Überwachung von DDL-Ereignissen und DML-Schreibvorgäng e
`dml_read, dml_write`	Wenn Sie diese Einstellung festlegen , wird das Auditing für

Wert	Beschreibung	
	alle DML-Ereignisse aktiviert	

> **Note**
>
> Eine Standardparametergruppe kann nicht abgeändert werden.

Weitere Informationen finden Sie hier:

- Amazon DocumentDB-Cluster-Parametergruppen erstellen

  Nach dem Erstellen einer Parametergruppe ändern Sie diese, indem Sie den `audit_logs`-Parameterwert auf `enabled` ändern.
- Amazon DocumentDB-Cluster-Parametergruppen ändern

Schritt 2. Aktivieren Sie den Amazon CloudWatch Logs-Export

Wenn der Wert des `audit_logs` Cluster-Parameters`enabled`,, oder `dml_write` ist `ddldml_read`, müssen Sie Amazon DocumentDB auch für den Export von Protokollen nach Amazon CloudWatch aktivieren. Wenn Sie einen dieser Schritte auslassen, werden keine Audit-Logs an gesendet. CloudWatch

Wenn Sie einen Cluster erstellen, einen Snapshot ausführen oder einen point-in-time-restore Snapshot wiederherstellen, können Sie CloudWatch Logs aktivieren, indem Sie die folgenden Schritte ausführen.

Using the AWS Management Console

Informationen zum Exportieren von Protokollen durch Amazon DocumentDB in die CloudWatch Konsole finden Sie in den folgenden Themen:

- Beim Erstellen eines Clusters — siehe Cluster erstellen: Zusätzliche Konfigurationen (Schritt 5, Protokollexporte) Erstellen eines Clusters und einer primären Instance mithilfe der AWS Management Console

- Beim Ändern eines vorhandenen Clusters — Ändern eines Amazon DocumentDB-Clusters
- Bei der Durchführung einer Cluster-Snapshot-Wiederherstellung — Wiederherstellen aus einem Cluster-Snapshot
- Bei der Durchführung einer point-in-time Wiederherstellung — Wiederherstellen auf einen bestimmten Zeitpunkt

Using the AWS CLI

So aktivieren Sie Prüfungsprotokolle beim Erstellen eines neuen Clusters

Der folgende Code erstellt den Cluster `sample-cluster` und aktiviert CloudWatch Audit-Logs.

Example

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
    --db-cluster-identifier sample-cluster \
    --port 27017 \
    --engine docdb \
    --master-username master-username \
    --master-user-password password \
    --db-subnet-group-name default \
    --enable-cloudwatch-logs-exports audit
```

Für Windows:

```
aws docdb create-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --port 27017 ^
    --engine docdb ^
    --master-username master-username ^
    --master-user-password password ^
    --db-subnet-group-name default ^
    --enable-cloudwatch-logs-exports audit
```

So aktivieren Sie Prüfungsprotokolle beim Ändern eines vorhandenen Clusters

Der folgende Code ändert den Cluster `sample-cluster` und aktiviert CloudWatch Auditprotokolle.

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Für Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Die Ausgabe dieser Operationen sieht in etwa wie folgt aus (JSON-Format).

```
{
    "DBCluster": {
        "HostedZoneId": "ZNKXH85TT8WVW",
        "StorageEncrypted": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "MasterUsername": "<user-name>",
        "BackupRetentionPeriod": 1,
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-cluster",
        "Status": "creating",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "DBSubnetGroup": "default",
        "DBClusterMembers": [],
```

```
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
        "DbClusterResourceId": "cluster-YOS52CUXGDTNKDQ7DH72I4LED4",
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "PreferredBackupWindow": "07:16-07:46",
        "DBClusterIdentifier": "sample-cluster"
    }
}
```

Deaktivieren des Prüfens

Sie können die Überwachung deaktivieren, indem Sie den CloudWatch Protokollexport und den Parameter deaktivieren. `audit_logs`

Protokollexport deaktivieren CloudWatch

Sie können den Export von Prüfprotokollen über die AWS Management Console oder AWS CLI deaktivieren.

Using the AWS Management Console

Im folgenden Verfahren wird der verwendetAWS Management Console, um den Export von Protokollen durch Amazon DocumentDB zu CloudWatch deaktivieren.

So deaktivieren Sie Prüfungsprotokolle

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster). Wählen Sie anschließend die Schaltfläche links neben dem Namen des Clusters aus, für den Sie das Exportieren von Protokollen deaktivieren möchten.
3. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.

4. Scrollen Sie nach unten zum Abschnitt Log exports (Protokollexporte) und wählen Sie Disabled (Deaktiviert) aus.
5. Klicken Sie auf Weiter.
6. Überprüfen Sie Ihre Änderungen. Wählen Sie anschließend den Zeitpunkt aus, an dem diese Änderung auf Ihren Cluster angewendet werden soll.
   - Apply during the next scheduled maintenance window (Anwendung während des nächsten geplanten Wartungsfensters)
   - Apply immediately (Sofort anwenden)
7. Wählen Sie Modify Cluster (Cluster bearbeiten).

Using the AWS CLI

Der folgende Code ändert den Cluster `sample-cluster` und deaktiviert CloudWatch Audit-Logs.

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

Für Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBCluster": {
        "DBClusterParameterGroup": "default.docdb4.0",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "MasterUsername": "<user-name>",
        "Status": "available",
        "Engine": "docdb",
        "Port": 27017,
```

```
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "EarliestRestorableTime": "2019-02-13T16:35:50.387Z",
        "DBSubnetGroup": "default",
        "LatestRestorableTime": "2019-02-13T16:35:50.387Z",
        "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-
cluster2",
        "Endpoint": "sample-cluster2.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster2.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "BackupRetentionPeriod": 1,
        "EngineVersion": "4.0.0",
        "MultiAZ": false,
        "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
        "DBClusterIdentifier": "sample-cluster2",
        "AssociatedRoles": [],
        "PreferredBackupWindow": "07:16-07:46",
        "DbClusterResourceId": "cluster-YOS52CUXGDTNKDQ7DH72I4LED4",
        "StorageEncrypted": false,
        "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

Deaktivieren des Parameters audit_logs

Um den Parameter `audit_logs` für Ihren Cluster zu deaktivieren, können Sie den Cluster so ändern, dass er eine Parametergruppe mit dem Wert `disabled` für den Parameter `audit_logs` verwendet. Sie können auch den Wert des Parameters `audit_logs` in der Parametergruppe des Clusters in `disabled` ändern.

Weitere Informationen finden Sie unter den folgenden Themen:

- Ändern eines Amazon DocumentDB-Clusters
- Amazon DocumentDB-Cluster-Parametergruppen ändern

Zugreifen auf Prüfereignisse

Gehen Sie wie folgt vor, um auf Ihre Prüfereignisse bei Amazon zuzugreifen. CloudWatch

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Stellen Sie sicher, dass Sie sich in derselben Region wie Ihr Amazon DocumentDB-Cluster befinden.
3. Wählen Sie im Navigationsbereich Logs (Logs) aus.
4. Um die Prüfprotokolle für Ihren Cluster zu finden, suchen Sie in der Liste und wählen Sie **`/aws/docdb/`*`yourClusterName`*`/audit`** aus.

   Die Prüfereignisse für Ihre Instances sind unter dem jeweiligen Instance-Namen verfügbar.

Sichern und Wiederherstellen in Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) sichert Ihre Daten kontinuierlich 1–35 Tage lang in Amazon Simple Storage Service (Amazon S3), sodass Sie schnell auf einen beliebigen Punkt innerhalb des Aufbewahrungszeitraums für Backups wiederherstellen können. Amazon DocumentDB erstellt im Rahmen dieses kontinuierlichen Backup-Prozesses auch automatische Snapshots Ihrer Daten.

> **Note**
>
> Dies sind serviceverwaltete Amazon S3-Buckets, und Sie haben keinen Zugriff auf die Sicherungsdateien. Wenn Sie Ihre eigenen Backups steuern möchten, folgen Sie den Anweisungen unter Dumping, Wiederherstellen, Importieren und Exportieren von Daten.

Sie können Backup-Daten auch über die Aufbewahrungsfrist hinaus aufbewahren, indem Sie einen manuellen Snapshot der Daten Ihres Clusters erstellen. Der Backup-Prozess hat keinen Einfluss auf die Leistung Ihres Clusters.

In diesem Abschnitt werden die Anwendungsfälle für die Backup-Funktionen in Amazon DocumentDB erläutert und Sie erfahren, wie Sie Backups für Ihre Amazon DocumentDB-Cluster verwalten.

Themen

Sichern und Wiederherstellen: Konzepte

Substantiv	Beschreibung	APIs (Verben)

Aufbewahrungszeitraum für Backups	Ein Zeitraum zwischen 1 und 35 Tagen, für den Sie eine point-in-time Wiederherstellung durchführen können.	`create-db-cluster`
		`modify-db-cluster`
		`restore-db-cluster-to-point-in-time`

Amazon DocumentDB-Speicher-Volume	Hochverfügbares und sehr robustes Speichervolume, das Daten auf sechs Arten in drei Availability Zones repliziert. Ein Amazon DocumentDB-Cluster ist	`create-db-cluster` `delete-db-cluster`

Substantiv	Beschreibung	APIs (Verben)
	unabhängig von der Anzahl der Instances im Cluster sehr langlebig.	

Backup-Fenster	Zeitspanne an dem Tag, an dem automatische Snapshots gemacht werden.	`create-db-cluster`
		`describe-db-cluster`
		`modify-db-cluster`

Substantiv	Beschreibung	APIs (Verben)

Automatischer Snapshot	Tägliche Snapshots, die vollständige Backups des Clusters sind und automatisch durch den kontinuierlichen Backup-Prozess in Amazon DocumentDB erstellt werden.	`restore-db-cluster-from-snapshot`
		
		`describe-db-cluster-snapshot-attributes`
		
		`describe-db-cluster-snapshots`

Substantiv	Beschreibung	APIs (Verben)

Manueller Snapshot	Snapshots, die Sie manuell erstellen, um Vollsicherungen eines Clusters über den Sicherungszeitraum hinaus aufzubewahren.	`create-db-cluster-snapshot`
		`copy-db-cluster-snapshot`
		`delete-db-cluster-snapshot`
		`describe-db-cluster-snapshot-attributes`
		`describe-db-cluster-snapshots`
		`modify-db-cluster-snapshot-attribute`

Grundlegendes zur Backup-Speicher-Nutzung

Der Amazon DocumentDB-Sicherungsspeicher besteht aus kontinuierlichen Sicherungen innerhalb des Aufbewahrungszeitraums für Sicherungen und manuellen Snapshots außerhalb des Aufbewahrungszeitraums. Wenn Sie Ihren Sicherungsspeicher kontrollieren möchten, können Sie das Aufbewahrungsintervall für Sicherungen verringern, alte (nicht mehr benötigte) manuelle Snapshots entfernen oder beides. Allgemeine Informationen zu Amazon DocumentDB-Backups finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB. Informationen zu den Preisen für Amazon DocumentDB-Sicherungsspeicher finden Sie unter Amazon DocumentDB-Preise.

Wenn Sie Ihre Kosten kontrollieren möchten, können Sie die Menge des Speichers überwachen, der nach Ablauf des Aufbewahrungszeitraums noch von kontinuierlichen Sicherungen und manuellen Snapshots belegt wird. Anschließend können Sie den Aufbewahrungszeitraum für Backups verringern und manuelle Snapshots entfernen, wenn sie nicht mehr benötigt werden.

Sie können die Amazon- CloudWatch Metriken `TotalBackupStorageBilled`, und verwenden`SnapshotStorageUsed`, `BackupRetentionPeriodStorageUsed` um die Menge des

von Ihren Amazon DocumentDB-Backups verwendeten Speichers wie folgt zu überprüfen und zu überwachen:

- `BackupRetentionPeriodStorageUsed` stellt den Umfang des Backup-Speichers zum Speichern von kontinuierlichen Backups zur aktuellen Zeit dar. Dieser Metrikwert hängt von der Größe des Cluster-Volumens und der Menge der von Ihnen während des Aufbewahrungszeitraums vorgenommenen Änderungen ab. Für Fakturierungszwecke übersteigt die Metrik die kumulative Cluster-Volume-Größe während des Aufbewahrungszeitraums nicht. Beispiel: Wenn Ihr Cluster 100 GiB groß ist und Ihr Aufbewahrungszeitraum zwei Tage beträgt, hat `BackRetentionPeriodStorageUsed` einen Maximalwert von 200 GiB (100 GiB+100 GiB).

- `SnapshotStorageUsed` stellt den Umfang des Backup-Speichers zum Speichern von manuellen Snapshots außerhalb des Aufbewahrungszeitraums für Backups dar. Innerhalb des Aufbewahrungszeitraums manuell aufgenommene Snapshots zählen nicht zu Ihrem Backup-Speicher. Ebenso werden automatische Snapshots nicht zu Ihrem Backup-Speicher gezählt. Die Größe einzelner Snapshots entspricht der Größe des Cluster-Volumes zum Zeitpunkt der Snapshot-Erstellung. Der `SnapshotStorageUsed`-Wert hängt von der Anzahl der Snapshots ab, die Sie aufbewahren, und von der Größe der einzelnen Snapshots. Beispiel: Angenommen, Sie haben einen Snapshot außerhalb des Aufbewahrungszeitraums und die Cluster-Volume-Größe betrug 100 GiB, als der Snapshot erstellt wurde. Dann beträgt die `SnapshotStorageUsed`-Menge 100 GiB.

- `TotalBackupStorageBilled` stellt die Summe von `BackupRetentionPeriodStorageUsed` und `SnapshotStorageUsed` abzüglich des freien Backup-Speichers gleich der Größe des Cluster-Volumes für einen Tag dar. Wenn Ihre Clustergröße beispielsweise 100 GiB beträgt, haben Sie einen Aufbewahrungstag und einen Snapshot außerhalb des Aufbewahrungszeitraums, der `TotalBackupStorageBilled` ist 100 GiB (100 GiB + 100 GiB – 100 GiB).

- Diese Metriken werden für jeden Amazon DocumentDB-Cluster unabhängig berechnet.

Sie können Ihre Amazon DocumentDB-Cluster überwachen und Berichte mithilfe CloudWatch von Metriken über die CloudWatch Konsole erstellen. Weitere Informationen zur Verwendung von CloudWatch Metriken finden Sie unter Amazon-Documententententent.

Dumping, Wiederherstellung, Import und Export von Daten

Sie können die `mongoimport` Dienstprogramme `mongodump`, `mongorestore`, und verwenden`mongoexport`, um Daten in und aus Ihrem Amazon DocumentDB-Cluster zu verschieben. In diesem Abschnitt wird der Zweck der einzelnen Tools und Konfigurationen beschrieben, damit Sie die Leistung verbessern können.

Themen

`mongodump`

Das Dienstprogramm `mongodump` erstellt eine binäre (BSON)-Sicherung einer MongoDB-Datenbank. Das `mongodump` Tool ist die bevorzugte Methode, um Daten aus Ihrer MongoDB-Quellbereitstellung zu dumpen, wenn Sie sie aufgrund der Größeneffizienzen, die durch das Speichern der Daten in einem Binärformat erzielt werden, in Ihrem Amazon DocumentDB-Cluster wiederherstellen möchten.

Abhängig von den Ressourcen, die auf der Instance oder dem Computer verfügbar sind, die bzw. den Sie zum Ausführen des Befehls verwenden, können Sie Ihre beschleunigen, `mongodump` indem Sie die Anzahl der parallelen Verbindungen erhöhen, die mit der `--numParallelCollections` Option von der Standard-1 entfernt wurden. Eine gute Faustregel besteht darin, mit einem Worker pro vCPU auf der primären Instance Ihres Amazon DocumentDB-Clusters zu beginnen.

> **Note**
>
> Wir empfehlen MongoDB-Datenbanktools bis einschließlich Version 100.6.1 für Amazon DocumentDB . Sie können hier auf die Downloads der MongoDB-Datenbanktools zugreifenhttps://www.mongodb.com/download-center/database-tools/releases/archive.

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des -mongodumpHilfsprogramms im Amazon DocumentDB-Cluster, `sample-cluster`.

```
mongodump --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --out=sample-output-file \
    --numParallelCollections 4  \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

mongorestore

Mit dem Dienstprogramm `mongorestore` können Sie eine Binärsicherung (BSON) einer Datenbank wiederherstellen, die mit dem Dienstprogramm `mongodump` erstellt wurde. Sie können die Wiederherstellungsleistung verbessern, indem Sie die Anzahl der Worker für jede Sammlung während der Wiederherstellung mit der Option `--numInsertionWorkersPerCollection` erhöhen. (Der Standardwert ist 1.) Eine gute Faustregel besteht darin, mit einem Worker pro vCPU auf der primären Instance Ihres Amazon DocumentDB-Clusters zu beginnen.

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des -mongorestoreHilfsprogramms im Amazon DocumentDB-Cluster, `sample-cluster`.

```
mongorestore --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem <fileToBeRestored>
```

mongoexport

Das `mongoexport` Tool exportiert Daten in Amazon DocumentDB in JSON-, CSV- oder TSV-Dateiformate. Das `mongoexport`-Tool ist die bevorzugte Methode für den Export von Daten, die für Menschen oder Maschinen lesbar sein müssen.

> **Note**
>
> `mongoexport` unterstützt parallele Exporte nicht direkt. Es ist jedoch möglich, die Leistung zu steigern, indem mehrere `mongoexport`-Aufgaben gleichzeitig für verschiedene Sammlungen ausgeführt werden.

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des -`mongoexport`Tools im Amazon DocumentDB-Cluster, `sample-cluster`.

```
mongoexport --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --out=sample-output-file \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

`mongoimport`

Das `mongoimport` Tool importiert den Inhalt von JSON-, CSV- oder TSV-Dateien in einen Amazon DocumentDB-Cluster. Mithilfe des Parameters `--numInsertionWorkers` können Sie den Import parallelisieren und beschleunigen. (Der Standardwert ist 1.)

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des -`mongoimport`Tools im Amazon DocumentDB-Cluster, `sample-cluster`.

```
mongoimport --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --file=<yourFile> \
    --numInsertionWorkers 4 \
    --username=sample-user \
```

```
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

Tutorial

Im folgenden Tutorial wird beschriebenmongodump, wie Sie die `mongoimport` Dienstprogramme , `mongoexport`, und verwendenmongorestore, um Daten in und aus einem Amazon DocumentDB-Cluster zu verschieben.

1. Voraussetzungen – Bevor Sie beginnen, stellen Sie sicher, dass Ihr Amazon DocumentDB-Cluster bereitgestellt ist und dass Sie Zugriff auf eine Amazon EC2 in derselben VPC wie Ihr Cluster haben. Weitere Informationen finden Sie unter Herstellen einer Verbindung über Amazon EC2.

   Um die mongo-Hilfstools verwenden zu können, muss das - mongodb-org-tools Paket wie folgt in Ihrer EC2-Instance installiert sein.

   ```
   sudo yum install mongodb-org-tools-4.0.18
   ```

   Da Amazon DocumentDB standardmäßig Transport Layer Security (TLS)-Verschlüsselung verwendet, müssen Sie auch die Amazon-RDS-Zertifizierungsstellendatei (CA) herunterladen, um die Mongo-Shell zum Herstellen einer Verbindung zu verwenden, und zwar wie folgt.

   ```
   wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
   ```

2. Herunterladen von Beispieldaten – Für dieses Tutorial laden Sie einige Beispieldaten herunter, die Informationen zu Bol enthalten.

   ```
   wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/
   datasets/restaurant.json
   ```

3. Importieren der Beispieldaten in Amazon DocumentDB – Da die Daten ein logisches JSON-Format haben, verwenden Sie das `mongoimport` Dienstprogramm , um die Daten in Ihren Amazon DocumentDB-Cluster zu importieren.

   ```
   mongoimport --ssl \
       --host="tutorialCluster.amazonaws.com:27017" \
       --collection=restaurants \
       --db=business \
       --file=restaurant.json \
   ```

```
    --numInsertionWorkers 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

4. Dump der Daten mit **`mongodump`** – Nachdem Sie nun Daten in Ihrem Amazon DocumentDB-Cluster haben, können Sie mit dem `mongodump` Dienstprogramm einen binären Dump dieser Daten erstellen.

```
mongodump --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017"\
    --collection=restaurants \
    --db=business \
    --out=restaurantDump.bson \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

5. **`restaurants`** Sammlung löschen – Bevor Sie die `restaurants` Sammlung in der `business` Datenbank wiederherstellen, müssen Sie zunächst die Sammlung, die bereits in dieser Datenbank vorhanden ist, wie folgt löschen.

```
use business
```

```
db.restaurants.drop()
```

6. Wiederherstellen der Daten mit **`mongorestore`** – Mit dem binären Dump der Daten aus Schritt 3 können Sie jetzt das `mongorestore` Dienstprogramm verwenden, um Ihre Daten in Ihrem Amazon DocumentDB-Cluster wiederherzustellen.

```
mongorestore --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem restaurantDump.bson
```

7. Exportieren der Daten mit **`mongoexport`** – Um das Tutorial abzuschließen, exportieren Sie die Daten aus Ihrem Cluster im Format einer JSON-Datei, nicht anders als die Datei, die Sie in Schritt 1 importiert haben.

```
mongoexport --ssl \
    --host="tutorialCluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=restaurants \
    --db=business \
    --out=restaurant2.json \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

8. Validierung – Mit den folgenden Befehlen können Sie überprüfen, ob die Ausgabe von Schritt 5 dasselbe Ergebnis wie Schritt 1 liefert.

```
wc -l restaurant.json
```

Ausgabe dieses Befehls:

```
2548 restaurant.json
```

```
wc -l restaurant2.json
```

Ausgabe dieses Befehls:

```
2548 restaurant2.json
```

Überlegungen zum Cluster-Snapshot

Amazon DocumentDB erstellt tägliche automatische Snapshots Ihres Clusters während des Backup-Fensters Ihres Clusters. Amazon DocumentDB speichert die automatischen Snapshots Ihres Clusters gemäß dem von Ihnen angegebenen Aufbewahrungszeitraum für Backups. Bei Bedarf können Sie Ihren Cluster zu einem beliebigen Zeitpunkt während der Aufbewahrungsdauer des Backups wiederherstellen. Automatische Snapshots treten nicht auf, während ein Kopiervorgang in derselben Region für denselben Cluster ausgeführt wird.

Themen

- Sicherungsspeicher
- Backup-Fenster

- Aufbewahrungszeitraum für Backups
- Kopieren der Cluster-Snapshot-Verschlüsselung

Zusätzlich zu den automatischen Cluster-Snapshots können Sie auch manuell einen Cluster-Snapshot erstellen. Sie können sowohl automatische als auch manuelle Snapshots kopieren. Weitere Informationen finden Sie unter Erstellen eines manuellen Cluster-Snapshots und Kopieren von Amazon DocumentDB-Cluster-Snapshots.

> **Note**
>
> Ihr Cluster muss sich im Status verfügbar befinden, damit ein automatischer Snapshot erstellt werden kann.
> Sie können einen automatisierten Amazon DocumentDB-Cluster-Snapshot nicht freigeben. Als Problemumgehung können Sie einen manuellen Snapshot erstellen, indem Sie den automatisierten Snapshot kopieren und dann diese Kopie freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Kopieren von Amazon DocumentDB-Cluster-Snapshots. Weitere Informationen zum Wiederherstellen eines Clusters aus einem Snapshot finden Sie unter Wiederherstellen aus einem Cluster-Snapshot.

Sicherungsspeicher

Ihr Amazon DocumentDB-Sicherungsspeicher für jede besteht aus dem Backup-Speicher, der für Ihren Aufbewahrungszeitraum für Backups erforderlich AWS-Region ist, der automatische und manuelle Cluster-Snapshots in dieser Region enthält. Der standardmäßige Aufbewahrungszeitraum für Backups beträgt 1 Tag. Weitere Informationen zu den Preisen für Backup-Speicher finden Sie unter Amazon DocumentDB – Preise.

Wenn Sie einen Cluster löschen, werden alle seine automatischen Snapshots gelöscht und können nicht wiederhergestellt werden. Manuelle Snapshots werden jedoch nicht gelöscht, wenn Sie einen Cluster löschen. Wenn Amazon DocumentDB einen endgültigen Snapshot (manueller Snapshot) erstellen soll, bevor Ihr Cluster gelöscht wird, können Sie den endgültigen Snapshot verwenden, um Ihren Cluster wiederherzustellen.

Weitere Informationen zu Snapshots und Speicher finden Sie unter Grundlegendes zur Backup-Speicher-Nutzung.

Backup-Fenster

Automatische Snapshots erfolgen täglich während des bevorzugten Backup-Fensters. Wenn der Snapshot mehr Zeit benötigt, als dem Backup-Fenster zugewiesen wurde, wird der Backup-Prozess bis zum Ende fortgesetzt, auch wenn das Backup-Fenster beendet ist. Das Backup-Fenster kann sich nicht mit dem wöchentlichen Wartungsfenster für den Cluster überschneiden.

Wenn Sie beim Erstellen des Clusters kein bevorzugtes Backup-Fenster angeben, weist Amazon DocumentDB ein standardmäßiges 30-minütiges Backup-Fenster zu. Dieses Fenster wird nach dem Zufallsprinzip aus einem 8-stündigen Zeitblock ausgewählt, der mit der Region Ihres Clusters verknüpft ist. Sie können Ihr bevorzugtes Backup-Fenster ändern, indem Sie den Cluster ändern. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Name der Region	Region	UTC-Zeitblock
USA Ost (Ohio)	us-east-2	03:00-11:00
USA Ost (Nord-Virginia)	us-east-1	03:00-11:00
USA West (Oregon)	us-west-2	06:00-14:00
Asien-Pazifik (Hongkong)	ap-east-1	06:00-14:00
Asien-Pazifik (Hyderabad)	ap-south-2	06:30–14:30
Asien-Pazifik (Mumbai)	ap-south-1	06:00-14:00
Asien-Pazifik (Seoul)	ap-northeast-2	13:00-21:00
Asien-Pazifik (Singapur)	ap-southeast-1	14:00–22:00 Uhr
Asien-Pazifik (Sydney)	ap-southeast-2	12:00–20:00 Uhr
Asien-Pazifik (Tokio)	ap-northeast-1	13:00-21:00
Kanada (Zentral)	ca-central-1	03:00-11:00
China (Beijing)	cn-north-1	06:00-14:00
China (Ningxia)	cn-northwest-1	06:00-14:00

Name der Region	Region	UTC-Zeitblock
Europa (Frankfurt)	eu-central-1	21:00–05:00
Europa (Irland)	eu-west-1	22:00-06:00
Europa (London)	eu-west-2	22:00-06:00
Europa (Mailand)	eu-south-1	02:00–10:00
Europa (Paris)	eu-west-3	23:59–07:29
South America (São Paulo)	sa-east-1	00:00–08:00
AWS GovCloud (USA-Ost)	us-gov-east-1	17:00–01:00
AWS GovCloud (USA-West)	us-gov-west-1	06:00-14:00

Aufbewahrungszeitraum für Backups

Der Aufbewahrungszeitraum für Backups ist die Anzahl der Tage, für die ein automatisches Backup aufbewahrt wird, bevor es automatisch gelöscht wird. Amazon DocumentDB unterstützt einen Aufbewahrungszeitraum für Backups von 1–35 Tagen.

Sie können die Aufbewahrungsdauer des Backups beim Anlegen eines Clusters festlegen. Wenn Sie die Aufbewahrungsdauer für das Backup nicht explizit festlegen, wird Ihrem Cluster die standardmäßige Aufbewahrungsdauer für das Backup von 1 Tag zugewiesen. Nachdem Sie einen Cluster erstellt haben, können Sie den Aufbewahrungszeitraum für Backups ändern, indem Sie den Cluster entweder über die AWS Management Console oder die ändern AWS CLI. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Kopieren der Cluster-Snapshot-Verschlüsselung

Die Cluster- und Snapshot-Verschlüsselung basiert auf einem KMS-Verschlüsselungsschlüssel. Die KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN), der KMS-Schlüsselbezeichner oder der KMS-Schlüsselalias für den KMS-Verschlüsselungsschlüssel.

Es gelten die folgenden Richtlinien und Einschränkungen:

- Die Verschlüsselung wird beim Erstellen eines Snapshots aus dem Cluster abgeleitet. Wenn der Cluster verschlüsselt ist, wird der Snapshot dieses Clusters mit demselben KMS-Schlüssel verschlüsselt. Wenn der Cluster nicht verschlüsselt ist, wird der Snapshot nicht verschlüsselt.
- Wenn Sie einen verschlüsselten Cluster-Snapshot aus Ihrem Amazon Web Services-Konto kopieren, können Sie einen Wert für angeben, `KmsKeyId` um die Kopie mit einem neuen KMS-Verschlüsselungsschlüssel zu verschlüsseln. Wenn Sie keinen Wert für angeben`KmsKeyId`, wird die Kopie des Cluster-Snapshots mit demselben KMS-Schlüssel wie der Quell-Cluster-Snapshot verschlüsselt.
- Wenn Sie einen verschlüsselten Cluster-Snapshot kopieren, der von einem anderen Amazon Web Services-Konto freigegeben wird, müssen Sie einen Wert für angeben`KmsKeyId`.
- Um einen verschlüsselten Cluster-Snapshot in eine andere Amazon Web Services-Region zu kopieren, setzen Sie `KmsKeyId` auf die KMS-Schlüssel-ID, die Sie zum Verschlüsseln der Kopie des Cluster-Snapshots in der Zielregion verwenden möchten. KMS-Verschlüsselungsschlüssel sind spezifisch für die Amazon Web Services-Region, in der sie erstellt werden, und Sie können keine Verschlüsselungsschlüssel aus einer Amazon Web Services-Region in einer anderen Amazon Web Services-Region verwenden.
- Wenn Sie einen unverschlüsselten Cluster-Snapshot kopieren und einen Wert für den `KmsKeyId` Parameter angeben, wird ein Fehler zurückgegeben.

Vergleich von automatischen und manuellen Snapshots

Im Folgenden finden Sie die wichtigsten Features von automatischen und manuellen Snapshots von Amazon DocumentDB (mit MongoDB-Kompatibilität).

Automatische Amazon DocumentDB-Snapshots verfügen über die folgenden wichtigen Funktionen:

- Automatische Snapshot-Benennung – Automatische Snapshot-Namen folgen dem Muster `rds:<cluster-name>-yyyy-mm-dd-hh-mm`, wobei das Datum und die Uhrzeit `yyyy-mm-dd-hh-mm` darstellt, zu der der Snapshot erstellt wurde.
- Wird automatisch nach einem Zeitplan erstellt – Wenn Sie einen Cluster erstellen oder ändern, können Sie den Aufbewahrungszeitraum für Backups auf einen Ganzzahlwert zwischen 1 und 35 Tagen festlegen. Standardmäßig haben neue Cluster eine Aufbewahrungsdauer von 1 Tag. Die Aufbewahrungsdauer für Backups definiert die Anzahl der Tage, die automatische Snapshots aufbewahrt werden, bevor sie automatisch gelöscht werden. Sie können automatische Backups auf Amazon DocumentDB-Clustern nicht deaktivieren.

Zusätzlich zur Einstellung der Aufbewahrungsdauer für das Backup legen Sie auch das Backup-Fenster fest (die Tageszeit, zu der automatische Snapshots erstellt werden).

- Löschen automatischer Snapshots – Automatische Snapshots werden gelöscht, wenn Sie den Cluster des automatischen Snapshots löschen. Sie können einen automatischen Snapshot nicht manuell löschen.
- Inkrementell – Während des Aufbewahrungszeitraums für Backups werden Datenbankaktualisierungen aufgezeichnet, sodass Änderungen inkrementell aufgezeichnet werden.
- Wiederherstellen aus einem automatischen Snapshot – Sie können mithilfe der AWS Management Console oder der aus einem automatischen Snapshot wiederherstellen AWS CLI. Wenn Sie mithilfe der aus einem Snapshot wiederherstellen AWS CLI, müssen Sie Instances separat hinzufügen, nachdem der Cluster verfügbar ist.
- Freigabe – Sie können einen automatisierten Cluster-Snapshot von Amazon DocumentDB nicht freigeben. Als Problemumgehung können Sie einen manuellen Snapshot erstellen, indem Sie den automatisierten Snapshot kopieren und dann diese Kopie freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Kopieren von Amazon DocumentDB-Cluster-Snapshots. Weitere Informationen zum Wiederherstellen eines Clusters aus einem Snapshot finden Sie unter Wiederherstellen aus einem Cluster-Snapshot.
- Sie können von jedem Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups aus wiederherstellen – Da Datenbankaktualisierungen inkrementell aufgezeichnet werden, können Sie Ihren Cluster zu einem beliebigen Zeitpunkt innerhalb des Aufbewahrungszeitraums für Backups wiederherstellen.

  Wenn Sie eine Wiederherstellung aus einem automatischen Snapshot oder aus einer point-in-time Wiederherstellung mit der durchführen AWS CLI, müssen Sie Instances separat hinzufügen, nachdem der Cluster verfügbar ist.

Manuelle Amazon DocumentDB-Snapshots verfügen über die folgenden wichtigen Funktionen:

- On-Demand erstellt – Manuelle Amazon DocumentDB-Snapshots werden On-Demand mit der Amazon DocumentDB-Managementkonsole oder erstellt AWS CLI.
- Löschen eines manuellen Snapshots – Ein manueller Snapshot wird nur gelöscht, wenn Sie ihn explizit entweder über die Amazon DocumentDB-Konsole oder löschen AWS CLI. Ein manueller Snapshot wird nicht gelöscht, wenn Sie seinen Cluster löschen.

- Vollständige Sicherungen – Wenn ein manueller Snapshot erstellt wird, wird eine vollständige Sicherung der Daten Ihres Clusters erstellt und gespeichert.
- Manuelle Snapshot-Benennung – Sie geben den Namen des manuellen Snapshots an. Amazon DocumentDB fügt dem Namen keinen `datetime` Zeitstempel hinzu, daher müssen Sie diese Informationen hinzufügen, wenn Sie sie in den Namen aufnehmen möchten.
- Wiederherstellen aus einem manuellen Snapshot – Sie können mithilfe der Konsole oder der aus einem manuellen Snapshot wiederherstellen AWS CLI. Wenn Sie mithilfe der aus einem Snapshot wiederherstellen AWS CLI, müssen Sie Instances separat hinzufügen, nachdem der Cluster verfügbar ist.
- Service Quotas – Sie sind auf maximal 100 manuelle Snapshots pro beschränkt AWS-Region.
- Freigabe – Sie können manuelle Cluster-Snapshots freigeben, die von autorisierten kopiert werden können AWS-Konten. Sie können verschlüsselte oder unverschlüsselte manuelle Snapshots freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Kopieren von Amazon DocumentDB-Cluster-Snapshots.
- Sie stellen auf wieder her, als der manuelle Snapshot erstellt wurde – Wenn Sie von einem manuellen Snapshot wiederherstellen, stellen Sie auf wieder her, als der manuelle Snapshot erstellt wurde.

  Wenn Sie mithilfe der aus einem Snapshot wiederherstellen AWS CLI, müssen Sie Instances separat hinzufügen, nachdem der Cluster verfügbar ist.

Erstellen eines manuellen Cluster-Snapshots

Sie können einen manuellen Snapshot entweder mit der AWS Management Console oder erstellen AWS CLI. Die Zeit, die für die Erstellung eines Snapshots benötigt wird, hängt von der Größe Ihrer Datenbanken ab. Wenn Sie einen Snapshot erstellen, müssen Sie Folgendes tun:

1. Identifizieren Sie den zu sichernden Cluster.
2. Geben Sie Ihrem Snapshot einen Namen. Dies ermöglicht es Ihnen, ihn später wiederherzustellen.

Using the AWS Management Console

Um einen manuellen Snapshot mit der zu erstellen AWS Management Console, können Sie eine der beiden folgenden Methoden anwenden.

1. Methode 1:

    1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
    2. Wählen Sie im Navigationsbereich die Option Snapshots.

        > **Tip**
        >
        > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

    3. Wählen Sie auf der Seite Snapshots die Option Create (Erstellen) aus.
    4. Auf der Seite Create cluster snapshot (Cluster-Snapshot erstellen):
        a. Cluster-ID – Wählen Sie aus der Dropdown-Liste der Cluster den Cluster aus, für den Sie einen Snapshot erstellen möchten.
        b. Snapshot-Kennung – Geben Sie einen Namen für Ihren Snapshot ein.

            Einschränkungen bei der Snapshot-Benennung:

            - Die Länge beträgt [1–255] Buchstaben, Zahlen oder Bindestriche.
            - Muss mit einem Buchstaben beginnen.
            - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
            - Muss für alle Cluster (über Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

        c. Wählen Sie Erstellen.

2. Methode 2:

    1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
    2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie auf der Seite Cluster die Schaltfläche links neben dem Cluster aus, den Sie als Snapshot aufnehmen möchten.
4. Wählen Sie im Menü Actions (Aktionen) die Option Take snapshot (Snapshot erstellen) aus.
5. Auf der Seite Create cluster snapshot (Cluster-Snapshot erstellen):
   a. Snapshot-Kennung – Geben Sie einen Namen für Ihren Snapshot ein.

      Einschränkungen bei der Snapshot-Benennung:

      - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Cluster (über Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

   b. Wählen Sie Erstellen.

Using the AWS CLI

Um einen Cluster-Snapshot mit der zu erstellen AWS CLI, verwenden Sie die `-create-db-cluster-snapshot`Operation mit den folgenden Parametern.

Parameter

- **`--db-cluster-identifier`** – Erforderlich. Der Name des Clusters, von dem Sie einen Snapshot machen. Dieser Cluster muss vorhanden sein und verfügbar sein.
- **`--db-cluster-snapshot-identifier`** – Erforderlich. Der Name des manuellen Snapshots, den Sie erstellen.

Im folgenden Beispiel wird ein Snapshot mit dem Namen `sample-cluster-snapshot` für einen Clusters namens `sample-cluster` erstellt.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Für Windows:

```
aws docdb create-db-cluster-snapshot ^
    --db-cluster-identifier sample-cluster ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-04-24T04:59:08.475Z",
        "Engine": "docdb",
        "Status": "creating",
        "Port": 0,
        "VpcId": "vpc-abc0123",
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:<accountID>:cluster-
snapshot:sample-cluster-snapshot"
    }
}
```

Kopieren von Amazon DocumentDB-Cluster-Snapshots

In Amazon DocumentDB können Sie manuelle und automatische Snapshots innerhalb derselben AWS-Region oder in eine andere AWS-Region innerhalb desselben Kontos kopieren. Sie können auch Snapshots freigeben, die anderen AWS-Konten in derselben gehören AWS-Region. Sie können einen Cluster-Snapshot jedoch nicht AWS-Konto in einem einzigen Schritt über hinweg AWS-Regionen und kopieren. Diese Aktionen müssen einzeln ausgeführt werden.

Alternativ zum Kopieren können Sie manuelle Snapshots auch mit anderen teilen AWS-Konten. Weitere Informationen finden Sie unter Freigeben von Amazon DocumentDB-Cluster-Snapshots.

> **Note**
>
> Amazon DocumentDB stellt Ihnen eine Rechnung basierend auf der Menge der von Ihnen aufbewahrten Sicherungs- und Snapshot-Daten und dem Aufbewahrungszeitraum in Rechnung. Weitere Informationen zum Speicher, der mit Amazon DocumentDB-Backups und -Snapshots verknüpft ist, finden Sie unter Grundlegendes zur Backup-Speicher-Nutzung. Informationen zu den Preisen für Amazon DocumentDB-Speicher finden Sie unter Amazon DocumentDB-Preise.

Themen

Kopieren freigegebener Snapshots

Sie können Snapshots kopieren, die von anderen für Sie freigegeben wurden AWS-Konten. Wenn Sie einen verschlüsselten Snapshot kopieren, der von einem anderen freigegeben wurde AWS-Konto, müssen Sie Zugriff auf den AWS KMS Verschlüsselungsschlüssel haben, der zum Verschlüsseln des Snapshots verwendet wurde.

Sie können einen freigegebenen Snapshot nur in dieselbe kopieren, unabhängig davon AWS-Region, ob der Snapshot verschlüsselt ist oder nicht. Weitere Informationen finden Sie unter Verschlüsselungen.

Kopieren von Snapshots über hinweg AWS-Regionen

Wenn Sie einen Snapshot in eine kopieren AWS-Region , die sich von der des Quell-Snapshots unterscheidet AWS-Region, ist jede Kopie ein vollständiger Snapshot. Eine vollständige Snapshot-Kopie enthält alle Daten und Metadaten, die für die Wiederherstellung des Amazon DocumentDB-Clusters erforderlich sind.

Abhängig von den AWS-Regionen beteiligten und der Menge der zu kopierenden Daten kann es Stunden dauern, bis eine regionsübergreifende Snapshot-Kopie abgeschlossen ist. In einigen Fällen kann es eine große Anzahl von regionsübergreifenden Snapshot-Kopieranforderungen aus einer bestimmten geben AWS-Region. In diesen Fällen stellt Amazon DocumentDB neue regionsübergreifende Kopieranforderungen dieser Quelle möglicherweise AWS-Region in eine Warteschlange, bis aktive Kopiervorgänge abgeschlossen sind. Zu Kopieranforderungen, die sich in der Warteschlange befinden, werden keine Fortschrittsinformationen angezeigt. Fortschrittsinformationen werden angezeigt, sobald der Kopiervorgang gestartet wird.

Einschränkungen

Im folgenden werden einige Einschränkungen beim Kopieren von Snapshots aufgeführt:

- Wenn Sie einen Quell-Snapshot löschen, bevor der Ziel-Snapshot verfügbar ist, kann das Kopieren des Snapshots fehlschlagen. Verifizieren Sie, dass der Ziel-Snapshot den Status `AVAILABLE` hat, bevor Sie einen Quell-Snapshot löschen.
- Pro Konto können bis zu fünf Snapshot-Kopieranforderungen an eine einzelne Zielregion aktiv sein.
- In Abhängigkeit von den beteiligten Regionen und der Menge der zu kopierenden Daten kann es Stunden dauern, bis eine regionsübergreifende Snapshot-Kopie fertiggestellt wird. Weitere Informationen finden Sie unter Kopieren von Snapshots über hinweg AWS-Regionen.

Verschlüsselungen

Sie können einen Snapshot kopieren, der mit einem AWS KMS -Verschlüsselungsschlüssel verschlüsselt wurde. Wenn Sie einen verschlüsselten Snapshot kopieren, muss auch die Kopie des Snapshots verschlüsselt werden. Wenn Sie einen verschlüsselten Snapshot

innerhalb derselben kopieren AWS-Region, können Sie die Kopie mit demselben AWS KMS Verschlüsselungsschlüssel wie den ursprünglichen Snapshot verschlüsseln oder einen anderen AWS KMS Verschlüsselungsschlüssel angeben. Wenn Sie einen verschlüsselten Snapshot regionsübergreifend kopieren, können Sie nicht denselben AWS KMS Verschlüsselungsschlüssel für die Kopie verwenden, der für den Quell-Snapshot verwendet wird, da die AWS KMS Schlüssel regionsspezifisch sind. Stattdessen müssen Sie einen - AWS KMS Schlüssel angeben, der in der Ziel- AWS-Regionn gültig ist.

Der Quell-Snapshot bleibt den gesamten Kopiervorgang über verschlüsselt. Weitere Informationen finden Sie unter Datenschutz in Amazon DocumentDB.

> **Note**
>
> Für Amazon DocumentDB-Cluster-Snapshots können Sie einen unverschlüsselten Cluster-Snapshot nicht verschlüsseln, wenn Sie den Snapshot kopieren.

Überlegungen zu Parametergruppen

Wenn Sie einen Snapshot regionsübergreifend kopieren, enthält die Kopie nicht die vom ursprünglichen Amazon DocumentDB-Cluster verwendete Parametergruppe. Wenn Sie einen Snapshot wiederherstellen, um einen neuen Cluster zu erstellen, erhält dieser Cluster die Standardparametergruppe für die , in der AWS-Region er erstellt wird. Um dem neuen Cluster dieselben Parameter wie dem Original zu geben, müssen Sie Folgendes tun:

1. AWS-Region Erstellen Sie in der Ziel- eine Amazon DocumentDB-Cluster-Parametergruppe mit denselben Einstellungen wie der ursprüngliche Cluster. Wenn bereits eine in der neuen vorhanden ist AWS-Region, können Sie diese verwenden.
2. Nachdem Sie den Snapshot in der Ziel- wiederhergestellt haben AWS-Region, ändern Sie den neuen Amazon DocumentDB-Cluster und fügen Sie die neue oder vorhandene Parametergruppe aus dem vorherigen Schritt hinzu. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Kopieren eines Cluster-Snapshots

Sie können einen Amazon DocumentDB-Cluster wie AWS CLIfolgt mit der AWS Management Console oder der kopieren.

Using the AWS Management Console

Führen Sie die folgenden Schritte aus AWS Management Console, um eine Kopie eines Cluster-Snapshots mit der zu erstellen. Dieses Verfahren funktioniert für das Kopieren verschlüsselter oder unverschlüsselter Cluster-Snapshots in derselben AWS-Region oder regionsübergreifend.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Snapshots und dann die Schaltfläche links neben dem Snapshot aus, den Sie kopieren möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Actions die Option Copy aus.
4. Füllen Sie auf der resultierenden Seite Cluster-Snapshot kopieren den Abschnitt Einstellungen aus.

   a. Zielregion – Optional. Um den Cluster-Snapshot in eine andere zu kopieren AWS-Region, wählen Sie diese AWS-Region für Zielregion aus.

   b. Neue Snapshot-ID – Geben Sie einen Namen für den neuen Snapshot ein.

      Einschränkungen bei der Benennung von Ziel-Snapshots:

      - Kann nicht der Namen eines vorhandenen Snapshots sein.
      - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.

   c. Tags kopieren – Um alle Tags, die Sie auf Ihrem Quell-Snapshot haben, in Ihre Snapshot-Kopie zu kopieren, wählen Sie Tags kopieren aus.

5. Füllen Sie den Abschnitt Encryption-at-rest aus.

   a. Verschlüsselung im Ruhezustand – Wenn Ihr Snapshot nicht verschlüsselt ist, stehen Ihnen diese Optionen nicht zur Verfügung, da Sie keine verschlüsselte Kopie aus einem unverschlüsselten Snapshot erstellen können. Wenn Ihr Snapshot verschlüsselt ist, können Sie die ändern, die während der Verschlüsselung im Ruhezustand AWS KMS key verwendet wird.

      Weitere Informationen zum Verschlüsseln von Snapshot-Kopien finden Sie unter Kopieren der Cluster-Snapshot-Verschlüsselung.

      Weitere Informationen zur Verschlüsselung im Ruhezustand finden Sie unter Verschlüsselung ruhender Amazon DocumentDB DocumentDB-Daten.

   b. Masterschlüssel – Wählen Sie aus der Dropdown-Liste eine der folgenden Optionen aus:

      - (Standard) aws/rds – Die Kontonummer und die AWS KMS Schlüssel-ID werden im Anschluss an diese Option aufgeführt.
      - **<some-key-name>** – Wenn Sie einen Schlüssel erstellt haben, wird dieser aufgelistet und steht Ihnen zur Auswahl.
      - Geben Sie einen Schlüssel-ARN ein – Geben Sie im Feld ARN den Amazon-Ressourcennamen (ARN) für Ihren AWS KMS Schlüssel ein. Das ARN-Format lautet `arn:aws:kms:<region>:<accountID>:key/<key-id>` .

6. Wenn Sie eine Kopie des ausgewählten Snapshots erstellen möchten, wählen Sie Copy Snapshot (Snapshot kopieren). Alternativ können Sie Abbrechen wählen, um keine Kopie des Snapshots zu erstellen.

Using the AWS CLI

Um eine Kopie eines unverschlüsselten Cluster-Snapshots mit der zu erstellen AWS CLI, verwenden Sie die `-copy-db-cluster-snapshot`Operation mit den folgenden Parametern. Wenn Sie den Snapshot in eine andere kopieren AWS-Region, führen Sie den Befehl in der aus AWS-Region , in die der Snapshot kopiert werden soll.

- **`--source-db-cluster-snapshot-identifier`** – Erforderlich. Die ID des zu kopierenden Cluster-Snapshots, der kopiert werden soll. Der Cluster-Snapshot muss vorhanden sein und muss sich im verfügbaren Zustand befindet. Wenn Sie den Snapshot in eine andere kopieren

AWS-Region, muss diese Kennung im ARN-Format für die Quell- vorliegen AWS-Region. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

- **`--target-db-cluster-snapshot-identifier`** – Erforderlich. Die ID des neuen Cluster-Snapshots, der aus dem Quell-Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

  Einschränkungen bei der Benennung von Ziel-Snapshots:

  - Kann nicht der Namen eines vorhandenen Snapshots sein.
  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
- **`--source-region`** – Wenn Sie den Snapshot in eine andere kopieren AWS-Region, geben Sie die an AWS-Region , aus der der verschlüsselte Cluster-Snapshot kopiert werden soll.

  Wenn Sie den Snapshot in ein anderes AWS-Region kopieren und nicht angeben`--source-region`, müssen Sie stattdessen die `pre-signed-url` Option angeben. Der `pre-signed-url` Wert muss eine URL sein, die eine mit Signature Version 4 signierte Anforderung für die `CopyDBClusterSnapshot` Aktion enthält, die in der Quelle aufgerufen werden soll AWS-Region , aus der der Cluster-Snapshot kopiert wird. Weitere Informationen zur finden `pre-signed-url`Sie unter CopyDBClusterSnapshot.
- **`--kms-key-id`** – Die KMS-Schlüsselkennung für den Schlüssel, der zum Verschlüsseln der Kopie des Cluster-Snapshots verwendet werden soll.

  Wenn Sie einen verschlüsselten Cluster-Snapshot in eine andere kopieren AWS-Region, ist dieser Parameter erforderlich. Sie müssen einen KMS-Schlüssel für die Ziel- angeben AWS-Region.

  Wenn Sie einen verschlüsselten Cluster-Snapshot in dieselbe kopieren AWS-Region, ist der AWS KMS Schlüsselparameter optional. Die Kopie des Cluster-Snapshots wird mit demselben AWS KMS Schlüssel wie der Quell-Cluster-Snapshot verschlüsselt. Wenn Sie einen neuen AWS KMS Verschlüsselungsschlüssel zum Verschlüsseln der Kopie angeben möchten, können Sie diesen Parameter verwenden.
- **`--copy-tags`** – Optional. Die Tags und Werte, die kopiert werden sollen.

Um einen Kopiervorgang abzubrechen, sobald er ausgeführt wird, können Sie den Ziel-Cluster-Snapshot löschen, der von oder identifiziert wurde, `--target-db-cluster-snapshot-identifier` `TargetDBClusterSnapshotIdentifier` während sich dieser Cluster-Snapshot im Kopierstatus befindet.

Example

Beispiel 1: Kopieren eines unverschlüsselten Snapshots in dieselbe Region

Im folgenden AWS CLI Beispiel wird eine Kopie von mit dem `sample-cluster-snapshot` Namen `sample-cluster-snapshot-copy` in derselben AWS-Region wie der Quell-Snapshot erstellt. Beim Erstellen der Kopie werden alle Tags des ursprünglichen Snapshots in die Snapshot-Kopie übernommen.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --copy-tags
```

Für Windows:

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
    --copy-tags
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-03-27T08:40:24.805Z",
        "Engine": "docdb",
        "Status": "copying",
```

```
        "Port": 0,
        "VpcId": "vpc-abcd0123",
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot"
    }
}
```

Example

Beispiel 2: Kopieren eines unverschlüsselten Snapshots über hinweg AWS-Regionen

Im folgenden AWS CLI Beispiel wird eine Kopie von mit `sample-cluster-snapshot`dem ARN erstellt`arn:aws:rds:us-east-1:123456789012:cluster-snapshot:sample-cluster-snapshot`. Diese Kopie heißt `sample-cluster-snapshot-copy` und befindet sich in der , AWS-Region in der der Befehl ausgeführt wird.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy
```

Für Windows:

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
```

```
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-04-29T16:45:51.239Z",
        "Engine": "docdb",
        "AllocatedStorage": 0,
        "Status": "copying",
        "Port": 0,
        "VpcId": "vpc-abc0123",
        "ClusterCreateTime": "2020-04-28T16:43:00.294Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "LicenseModel": "docdb",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": false,
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot",
    }
}
```

Example

Beispiel 3: Kopieren eines verschlüsselten Snapshots über hinweg AWS-Regionen

Im folgenden AWS CLI Beispiel wird eine Kopie von `sample-cluster-snapshot` aus der Region us-west-2 in die Region us-east-1 erstellt. Dieser Befehl wird in der Region us-east-1 aufgerufen.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --source-region us-west-2 \
```

```
    --kms-key-id sample-us-east-1-key
```

Für Windows:

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
    --source-region us-west-2 ^
    --kms-key-id sample-us-east-1-key
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "ayhu-xrsc-test-ap-southeast-1-small-cluster-kms",
        "SnapshotCreateTime": "2020-04-29T16:45:53.159Z",
        "Engine": "docdb",
        "AllocatedStorage": 0,
        "Status": "copying",
        "Port": 0,
        "ClusterCreateTime": "2020-04-28T16:43:07.129Z",
        "MasterUsername": "chimera",
        "EngineVersion": "4.0.0",
        "LicenseModel": "docdb",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-west-2:111122223333:cluster-
snapshot:sample-cluster-snapshot",
    }
}
```

> **Note**
>
> Weitere Informationen zum Verschlüsseln von Snapshot-Kopien finden Sie unter Kopieren der Cluster-Snapshot-Verschlüsselung.

Weitere Informationen zur Verschlüsselung im Ruhezustand finden Sie unter Verschlüsselung ruhender Amazon DocumentDB DocumentDB-Daten.

Freigeben von Amazon DocumentDB-Cluster-Snapshots

In Amazon DocumentDB können Sie manuelle Cluster-Snapshots freigeben, die von autorisierten kopiert werden können AWS-Konten. Sie können verschlüsselte oder unverschlüsselte manuelle Snapshots freigeben. Wenn Sie einen unverschlüsselten Snapshot freigeben, AWS-Konten kann der autorisierte den Cluster direkt aus dem Snapshot wiederherstellen, anstatt eine Kopie davon zu erstellen und daraus wiederherzustellen. Allerdings können Sie einen Cluster nicht aus einem Snapshot wiederherstellen, der sowohl freigegeben als auch verschlüsselt ist. Stattdessen können Sie eine Kopie des Clusters erstellen und den Cluster aus dieser Kopie wiederherstellen. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Kopieren von Amazon DocumentDB-Cluster-Snapshots.

Note

Sie können einen automatisierten Amazon DocumentDB-Cluster-Snapshot nicht freigeben. Als Problemumgehung können Sie einen manuellen Snapshot erstellen, indem Sie den automatisierten Snapshot kopieren und dann diese Kopie freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Kopieren von Amazon DocumentDB-Cluster-Snapshots. Weitere Informationen zum Wiederherstellen eines Clusters aus einem Snapshot finden Sie unter Wiederherstellen aus einem Cluster-Snapshot.

Sie können einen manuellen Snapshot für bis zu 20 andere freigeben AWS-Konten. Sie können darüber hinaus einen nicht verschlüsselten Snapshot als öffentlich freigeben. Damit ist der Snapshot für alle -Konten verfügbar. Vergewissern Sie sich bei der Freigabe eines Snapshots als öffentlich darauf, dass in Ihren öffentlichen Snapshots keine privaten Informationen enthalten sind.

Wenn Sie manuelle Snapshots mit anderen teilen AWS-Kontenund einen Cluster aus einem freigegebenen Snapshot mithilfe der AWS CLI oder der Amazon DocumentDB-API wiederherstellen, müssen Sie den Amazon-Ressourcennamen (ARN) des freigegebenen Snapshots als Snapshot-ID angeben.

Freigeben eines verschlüsselten Snapshots

Die folgenden Einschränkungen gelten für die Freigabe verschlüsselter Snapshots:

- Sie können verschlüsselte Snapshots nicht als öffentlich freigeben.
- Sie können keinen Snapshot freigeben, der mit dem AWS KMS Standardverschlüsselungsschlüssel des Kontos verschlüsselt wurde, das den Snapshot freigegeben hat.

Führen Sie die folgenden Schritte aus, um verschlüsselte Snapshots freizugeben.

1. Geben Sie den AWS Key Management Service (AWS KMS)-Verschlüsselungsschlüssel, der zum Verschlüsseln des Snapshots verwendet wurde, für alle Konten frei, die auf den Snapshot zugreifen können sollen.

   Sie können AWS KMS Verschlüsselungsschlüssel für andere AWS Konten freigeben, indem Sie die anderen Konten der AWS KMS Schlüsselrichtlinie hinzufügen. Weitere Informationen zum Aktualisieren einer Schlüsselrichtlinie finden Sie unter Verwenden von Schlüsselrichtlinien in AWS KMS im AWS Key Management Service Entwicklerhandbuch für . Ein Beispiel für die Erstellung einer Schlüsselrichtlinie finden Sie unter Erstellen einer IAM-Richtlinie, um das Kopieren des verschlüsselten Snapshots zu ermöglichen an späterer Stelle in diesem Thema.
2. Verwenden Sie die AWS CLI, wie unten gezeigt, um den verschlüsselten Snapshot für die anderen Konten freizugeben.

Gewähren des Zugriffs auf einen AWS KMS Verschlüsselungsschlüssel

Damit ein anderes einen verschlüsselten Snapshot kopieren AWS-Konto kann, der von Ihrem Konto freigegeben wurde, muss das Konto, für das Sie Ihren Snapshot freigeben, Zugriff auf den AWS KMS Schlüssel haben, der den Snapshot verschlüsselt hat. Um einem anderen Konto Zugriff auf einen - AWS KMS Schlüssel zu gewähren, aktualisieren Sie die Schlüsselrichtlinie für den AWS KMS Schlüssel mit dem ARN des Kontos, für das Sie als Prinzipal in der AWS KMS Schlüsselrichtlinie freigeben. Lassen Sie anschließend die Aktion `kms:CreateGrant` zu.

Nachdem Sie einem Konto Zugriff auf Ihren AWS KMS Verschlüsselungsschlüssel gewährt haben, muss dieses Konto zum Kopieren Ihres verschlüsselten Snapshots einen AWS Identity and Access Management (IAM)-Benutzer erstellen, falls es noch keinen hat. Darüber hinaus muss dieses Konto diesem IAM-Benutzer auch eine IAM-Richtlinie anfügen, die es dem Benutzer ermöglicht, einen

verschlüsselten Snapshot mit Ihrem AWS KMS Schlüssel zu kopieren. Das Konto muss ein IAM-Benutzer sein und darf aufgrund von AWS KMS Sicherheitseinschränkungen keine AWS-Konto Stammidentität sein.

Im folgenden Beispiel für eine Schlüsselrichtlinie ist der Benutzer 123451234512 der Eigentümer des AWS KMS Verschlüsselungsschlüssels. Bei Benutzer 123456789012 handelt es sich um das Konto, für das der Schlüssel freigegeben wird. Diese aktualisierte Schlüsselrichtlinie gewährt dem Konto Zugriff auf den AWS KMS Schlüssel. Dazu wird der ARN für die AWS-Konto Stammidentität für den Benutzer 123456789012 als Prinzipal für die Richtlinie aufgenommen und die `kms:CreateGrant` Aktion zugelassen.

```
{
    "Id": "key-policy-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::123451234512:user/KeyUser",
                "arn:aws:iam::123456789012:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"},
            {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::123451234512:user/KeyUser",
                "arn:aws:iam::123456789012:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
```

```
            ],
            "Resource": "*",
            "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
        }
    ]
}
```

Erstellen einer IAM-Richtlinie, um das Kopieren des verschlüsselten Snapshots zu ermöglichen

Wenn der externe Zugriff auf Ihren AWS KMS Schlüssel AWS-Konto hat, kann der Besitzer dieses Kontos eine Richtlinie erstellen, die es einem für das Konto erstellten IAM-Benutzer ermöglicht, einen verschlüsselten Snapshot zu kopieren, der mit diesem AWS KMS Schlüssel verschlüsselt ist.

Das folgende Beispiel zeigt eine Richtlinie, die einem IAM-Benutzer für AWS-Konto 123456789012 angefügt werden kann. Die Richtlinie ermöglicht es dem IAM-Benutzer, einen freigegebenen Snapshot aus Konto 123451234512 zu kopieren, der mit dem AWS KMS Schlüssel `c989c1dd-a3f2-4a5d-8d96-e793d082ab26` in der Region us-west-2 verschlüsselt wurde.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUseOfTheKey",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"]
        },
        {
            "Sid": "AllowAttachmentOfPersistentResources",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
```

```
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

Weitere Informationen zum Aktualisieren einer Schlüsselrichtlinie finden Sie unter Verwenden von Schlüsselrichtlinien in AWS KMS im AWS Key Management Service Entwicklerhandbuch für .

Freigeben eines Snapshots

Um einen Snapshot freizugeben, verwenden Sie die Amazon DocumentDB-`modify-db-snapshot-attribute`Operation. Verwenden Sie den `--values-to-add`Parameter, um eine Liste der IDs für das hinzuzufügen AWS-Konten , die zur Wiederherstellung des manuellen Snapshots autorisiert sind.

Im folgenden Beispiel können zwei AWS-Konto Kennungen, 123451234512 und 123456789012, den Snapshot mit dem Namen wiederherstellen`manual-snapshot1`. Außerdem wird der `all`-Attributwert entfernt, um den Snapshot als privat zu markieren.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add '["123451234512","123456789012"]'
```

Für Windows:

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-add '["123451234512","123456789012"]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512",
                    "123456789012"
                ]
            }
        ]
    }
}
```

Verwenden Sie den `--values-to-remove` Parameter , um eine AWS-Konto Kennung aus der Liste zu entfernen. Das folgende Beispiel verhindert, dass die AWS-Konto ID 123456789012 den Snapshot wiederherstellt.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove '["123456789012"]'
```

Für Windows:

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-remove '["123456789012"]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
```

```
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512"
                ]
            }
        ]
    }
}
```

Wiederherstellen aus einem Cluster-Snapshot

Amazon DocumentDB (mit MongoDB-Kompatibilität) erstellt einen Cluster-Snapshot Ihres Speicher-Volumes. Sie können einen neuen Cluster erstellen, indem Sie ihn aus einem Cluster-Snapshot wiederherstellen. Wenn Sie den Cluster wiederherstellen, geben Sie den Namen des wiederherzustellenden Cluster-Snapshots und einen Namen für den neuen Cluster an, der durch die Wiederherstellung erstellt wird. Sie können nicht von einem Snapshot auf einen bestehenden Cluster wiederherstellen, da bei der Wiederherstellung ein neuer Cluster erstellt wird.

Wenn Sie einen Cluster aus einem Cluster-Snapshot wiederherstellen:

- Diese Aktion stellt nur den Cluster wieder her, nicht die Instances für diesen Cluster. Sie müssen die Aktion `create-db-instance` aufrufen, um Instances für den wiederhergestellten Cluster zu erstellen, wobei Sie in `--db-cluster-identifier` die ID des wiederhergestellten Clusters angeben. Sie können Instances erst erstellen, nachdem der Cluster den Status available hat.
- Sie können einen verschlüsselten Snapshot nicht in einem unverschlüsselten Cluster wiederherstellen. Sie können jedoch einen unverschlüsselten Snapshot in einem verschlüsselten Cluster wiederherstellen, indem Sie den AWS KMS Schlüssel angeben.
- Um einen Cluster aus einem verschlüsselten Snapshot wiederherzustellen, müssen Sie Zugriff auf den AWS KMS Schlüssel haben.

> **Note**
>
> Sie können einen 3.6-Cluster nicht auf einem 4.0-Cluster wiederherstellen, aber Sie können von einer Clusterversion zu einer anderen migrieren. Weitere Informationen finden Sie unter Migration zu Amazon DocumentDB.

Using the AWS Management Console

Das folgende Verfahren zeigt, wie Sie einen Amazon DocumentDB-Cluster mithilfe der Amazon DocumentDB-Managementkonsole aus einem Cluster-Snapshot wiederherstellen.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Snapshots und dann die Schaltfläche links neben dem Snapshot aus, mit der Sie einen Cluster wiederherstellen möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Aktionen die Option Restore (Wiederherstellen).
4. Füllen Sie auf der Seite Restore snapshot (Snapshot wiederherstellen) den Abschnitt Configuration (Konfiguration) aus.

   a. Cluster-ID – Der Name für den neuen Cluster. Sie können den von Amazon DocumentDB bereitgestellten Namen akzeptieren oder einen Namen eingeben, den Sie bevorzugen. Der von Amazon DocumentDBsupplied Name hat das Format `docdb-` plus einen UTC-Zeitstempel, z. B. `docdb-`*`yyyy-mm-dd-hh-mm-ss`*.

   b. Instance-Klasse – Die Instance-Klasse für den neuen Cluster. Sie können die Standard-Instance-Klasse akzeptieren oder eine Instance-Klasse aus der Dropdown-Liste auswählen.

   c. Anzahl der Instances – Die Anzahl der Instances, die Sie mit diesem Cluster erstellen möchten. Sie können die Standardeinstellung von 3 Instances akzeptieren (1 primäre Instance mit Lese-/Schreibzugriff und 2 schreibgeschützte Replikate) oder die Anzahl der Instances aus der Dropdown-Liste auswählen.

5. Wählen Sie für Cluster-Speicherkonfiguration eine Speicheroption aus.

> **Note**
>
> Amazon DocumentDB-E/A-optimierte Speicherkonfiguration ist nur in der Amazon DocumentDB-5.0-Engine-Version verfügbar.

6. Wenn Sie mit der Cluster-Konfiguration zufrieden sind, wählen Sie Restore cluster (Cluster wiederherstellen) aus und warten Sie, bis Ihr Cluster wiederhergestellt ist.
7. Wenn Sie lieber einige Konfigurationen ändern möchten, z. B. die Angabe einer nicht standardmäßigen Amazon VPC oder Sicherheitsgruppe, wählen Sie unten links auf der Seite Erweiterte Einstellungen anzeigen aus und fahren Sie dann mit den folgenden Schritten fort.

   a. Vervollständigen Sie den Bereich Network settings (Netzwerkeinstellungen).

      - Virtual Private Cloud (VPC) – Akzeptieren Sie die aktuelle VPC oder wählen Sie eine VPC aus der Dropdown-Liste aus.
      - Subnetzgruppe – Akzeptieren Sie die `default` Subnetzgruppe oder wählen Sie eine aus der Dropdown-Liste aus.
      - VPC-Sicherheitsgruppen – Akzeptieren Sie die `default (VPC)` Sicherheitsgruppe oder wählen Sie eine aus der Liste aus.

   b. Vervollständigen Sie den Abschnitt Cluster options (Cluster-Optionen).

      - Datenbankport – Akzeptieren Sie den Standardport, oder verwenden Sie den Aufwärts- oder Abwärtspfeil`27017`, um den Port festzulegen, den Sie für Anwendungsverbindungen verwenden möchten.

   c. Vervollständigen Sie den Abschnitt Encryption (Verschlüsselung).

      - Verschlüsselung im Ruhezustand – Wenn Ihr Snapshot verschlüsselt ist, stehen Ihnen diese Optionen nicht zur Verfügung. Wenn er nicht verschlüsselt ist, können Sie eine der folgenden Optionen auswählen:
        - Um alle Daten Ihres Clusters zu verschlüsseln, wählen Sie Aktivieren aus encryption-at-rest. Wenn Sie diese Option auswählen, müssen Sie einen Master key (Hauptschlüssel) angeben.
        - Um die Daten Ihres Clusters nicht zu verschlüsseln, wählen Sie Deaktivieren aus encryption-at-rest. Wenn Sie diese Option wählen, sind Sie mit dem Abschnitt "Verschlüsselung" fertig.

- Masterschlüssel – Wählen Sie eine der folgenden Optionen aus der Dropdown-Liste aus:
  - (Standard) aws/rds – Die Kontonummer und die AWS KMS Schlüssel-ID werden im Anschluss an diese Option aufgeführt.
  - Vom Kunden verwalteter Schlüssel – Diese Option ist nur verfügbar, wenn Sie einen IAM-Verschlüsselungsschlüssel in der AWS Identity and Access Management (IAM)-Konsole erstellt haben. Sie können den Schlüssel für die Verschlüsselung Ihres Clusters auswählen.
  - Geben Sie einen Schlüssel-ARN ein – Geben Sie im Feld ARN den Amazon-Ressourcennamen (ARN) für Ihren AWS KMS Schlüssel ein. Das ARN-Format lautet `arn:aws:kms:<region>:<accountID>:key/<key-id>`.

d. Füllen sie den Abschnitt Log exports (Protokollexporte) aus.

- Wählen Sie die Protokolltypen aus, in denen veröffentlicht werden soll CloudWatch – Wählen Sie eine der folgenden Optionen aus:
  - Aktiviert – Ermöglicht Ihrem Cluster, die DDL-Protokollierung nach Amazon CloudWatch Logs zu exportieren.
  - Deaktiviert – Verhindert, dass Ihr Cluster DDL-Protokolle nach Amazon CloudWatch Logs exportiert. Disabled (Deaktiviert) ist die Voreinstellung.
- IAM-Rolle – Wählen Sie in der Liste die Option RDS-Service-verknüpfte Rolle aus.

e. Füllen Sie den Abschnitt Tags aus.

- Tag hinzufügen – Geben Sie im Feld Schlüssel den Namen für das Tag für Ihren Cluster ein. Geben Sie optional im Feld Value (Wert) den Tag-Wert ein. Tags werden mit AWS Identity and Access Management (IAM)-Richtlinien verwendet, um den Zugriff auf Amazon DocumentDB-Ressourcen zu verwalten und zu steuern, welche Aktionen auf die Ressourcen angewendet werden können.

f. Füllen Sie den Abschnitt Deletion protection (Löschschutz) aus.

- Aktivieren des Löschschutzes – Schutz den Cluster vor versehentlichem Löschen. Wenn diese Option aktiviert ist, können Sie den Cluster nicht löschen.

8. Wählen Sie Restore cluster (Cluster wiederherstellen) aus.

Using the AWS CLI

Um einen Cluster mithilfe der aus einem Snapshot wiederherzustellen AWS CLI, verwenden Sie die `-restore-db-cluster-from-snapshot`Operation mit den folgenden Parametern. Weitere Informationen finden Sie unter RestoreDBClusterFromSnapshot.

- **`--db-cluster-identifier`** – Erforderlich. Der Name des Clusters, der von der Operation erstellt wird. Ein Cluster mit diesem Namen darf vor dieser Operation nicht vorhanden sein.

  Einschränkungen bei der Benennung von Clustern:

  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
- **`--snapshot-identifier`** – Erforderlich. Der Name des Snapshots, der für die Wiederherstellung verwendet wird. Ein Snapshot mit diesem Namen muss vorhanden sein und sich im verfügbaren Zustand befindet.
- **`--engine`** – Erforderlich. Der Wert muss `docdb` sein.
- **`--storage-type standard | iopt1`** – Optional. Standard: `standard`.
- **`--kms-key-id`** – Optional. Der ARN der AWS KMS Schlüsselkennung, die beim Wiederherstellen eines verschlüsselten Snapshots oder beim Verschlüsseln eines Clusters beim Wiederherstellen aus einem unverschlüsselten Snapshot verwendet werden soll. Die Angabe der AWS KMS Schlüssel-ID führt dazu, dass der wiederhergestellte Cluster mit dem AWS KMS Schlüssel verschlüsselt wird, unabhängig davon, ob der Snapshot verschlüsselt wurde oder nicht.

  Das Format der `--kms-key-id` ist `arn:aws:kms:<region>:<accountID>:key/<key-id>`. Wenn Sie keinen Wert für den `--kms-key-id`-Parameter angeben, geschieht folgendes:

  - Wenn der Snapshot in verschlüsselt `--snapshot-identifier` ist, wird der wiederhergestellte Cluster mit demselben AWS KMS Schlüssel verschlüsselt, der zur Verschlüsselung des Snapshots verwendet wurde.
  - Wenn der Snapshot in `--snapshot-identifier` nicht verschlüsselt ist, dann ist der wiederhergestellte Cluster nicht verschlüsselt.

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-from-snapshot \
    --db-cluster-identifier sample-cluster-restore \
    --snapshot-identifier sample-cluster-snapshot \
    --engine docdb \
    --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

Für Windows:

```
aws docdb restore-db-cluster-from-snapshot ^
    --db-cluster-identifier sample-cluster-restore ^
    --snapshot-identifier sample-cluster-snapshot ^
    --engine docdb ^
    --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster-restore",
        "DBClusterParameterGroup": "default.docdb4.0",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "Port": 27017,
        "MasterUsername": "<master-user>",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [],
```

```
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcdefgh",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-
restore",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-04-01T01:43:40.871Z",
        "DeletionProtection": true
    }
}
```

Nachdem der Clusterstatus available lautet, erstellen Sie mindestens eine Instance für den Cluster.

Für Linux, macOS oder Unix:

```
aws docdb create-db-instance \
    --db-cluster-identifier sample-cluster-restore  \
    --db-instance-identifier sample-cluster-restore-instance \
    --availability-zone us-east-1b \
    --promotion-tier 2 \
    --db-instance-class db.r5.large \
    --engine docdb
```

Für Windows:

```
aws docdb create-db-instance ^
    --db-cluster-identifier sample-cluster-restore  ^
    --db-instance-identifier sample-cluster-restore-instance ^
    --availability-zone us-east-1b ^
    --promotion-tier 2 ^
    --db-instance-class db.r5.large ^
    --engine docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-cluster-restore-instance",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "creating",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcdefgh",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-west-2b",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcdefgh",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    ...
                }
            ]
        },
        "PreferredMaintenanceWindow": "fri:09:43-fri:10:13",
        "PendingModifiedValues": {},
        "EngineVersion": "4.0.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster-restore",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
```

```
        "PromotionTier": 2,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-cluster-
restore-instance"
    }
}
```

Wiederherstellen auf einen bestimmten Zeitpunkt

Sie können einen Cluster zu jedem beliebigen Zeitpunkt wiederherstellen, der innerhalb des Aufbewahrungszeitraums für Backups des Clusters liegt, indem Sie die AWS Management Console oder AWS Command Line Interface die (AWS CLI) verwenden.

> **Note**
>
> Sie können keine point-in-time Wiederherstellung eines 3.6-Clusters auf einen 4.0-Cluster durchführen, aber Sie können von einer Clusterversion zu einer anderen migrieren. Weitere Informationen finden Sie unter Migration zu Amazon DocumentDB.

Beachten Sie Folgendes, wenn Sie einen Cluster zu einem bestimmten Zeitpunkt wiederherstellen.

- Der neue Cluster wird mit der gleichen Konfiguration wie der Quell-Cluster erstellt, nur dass der neue Cluster mit der Standard-Parametergruppe erstellt wird. Um die Parametergruppe des neuen Clusters auf die Parametergruppe des Quellclusters festzulegen, ändern Sie den Cluster, nachdem er den Status available (verfügbar) hat. Weitere Informationen zum Ändern eines Clusters finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Using the AWS Management Console

Sie können einen Cluster point-in-time innerhalb seines Aufbewahrungszeitraums für Backups in einem wiederherstellen, indem Sie Folgendes mithilfe der abschließen AWS Management Console.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Cluster, den Sie wiederherstellen möchten.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Actions (Aktionen) die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen).
4. Geben Sie im Bereich Restore time (Wiederherstellungszeit) Datum und Uhrzeit für die zeitpunktbezogene Wiederherstellung an.

   a. Wiederherstellungsdatum – Wählen Sie ein Datum aus, das zwischen der frühesten und der letzten Wiederherstellungszeit liegt, oder geben Sie es ein.

   b. Wiederherstellungszeit – Wählen Sie die Stunde, Minute und Sekunden aus, die zwischen der frühesten und der letzten Wiederherstellungszeit liegen, oder geben Sie sie ein.

5. Füllen Sie den Bereich Configuration (Konfiguration) aus.

   a. Cluster-ID – Akzeptieren Sie die Standard-ID oder geben Sie eine ID ein, die Sie bevorzugen.

      Einschränkungen bei der Benennung von Clustern:

      - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.

   b. Instance-Klasse – Wählen Sie aus der Dropdown-Liste die gewünschte Instance-Klasse für die Instances des Clusters aus.

   c. Anzahl der Instances – Wählen Sie aus der Dropdown-Liste die Anzahl der Instances aus, die bei der Wiederherstellung des Clusters erstellt werden sollen.

6. Wählen Sie für Cluster-Speicherkonfiguration eine Speicheroption aus.

> **Note**
>
> Amazon DocumentDB-E/A-optimierte Speicherkonfiguration ist nur in der Amazon DocumentDB-5.0-Engine-Version verfügbar.

7. Optional. Um die Netzwerkeinstellungen, Cluster-Optionen und die Aktivierung von Protokollexporten zu konfigurieren, wählen Sie Show advanced settings (Erweiterte Einstellungen anzeigen) aus. Schließen Sie dann die folgenden Abschnitte ab. Fahren Sie andernfalls mit dem nächsten Schritt fort.

   - Network settings (Netzwerkeinstellungen)

     1. Virtual Private Cloud (VPC) – Wählen Sie aus der Dropdown-Liste die VPC aus, die Sie für diesen Cluster verwenden möchten.
     2. Subnetzgruppe – Wählen Sie aus der Dropdown-Liste die Subnetzgruppe für diesen Cluster aus.
     3. VPC-Sicherheitsgruppen – Wählen Sie aus der Dropdown-Liste die VPC-Sicherheitsgruppen für diesen Cluster aus.

   - Cluster options (Cluster-Optionen)

     1. Port – Akzeptieren Sie den Standardport (27017) oder verwenden Sie die Aufwärts- und Abwärtspfeile, um den Port für die Kommunikation mit diesem Cluster festzulegen.

   - Protokollexporte

     1. Prüfungsprotokolle – Wählen Sie diese Option aus, um den Export von Prüfungsprotokollen nach Amazon CloudWatch Logs zu aktivieren. Wenn Sie diese Option auswählen, müssen Sie `audit_logs` in der benutzerdefinierten Parametergruppe des Clusters aktivieren. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.
     2. Profiler-Protokolle – Wählen Sie diese Option aus, um den Export von Operationsprofiler-Protokollen nach Amazon CloudWatch Logs zu aktivieren. Wenn Sie diese Option auswählen, müssen Sie auch die folgenden Parameter in der benutzerdefinierten Parametergruppe des Clusters ändern:
        - `profiler` – Setzen Sie auf `enabled`.

- `profiler_threshold_ms` – Legen Sie auf einen Wert fest`[0-INT_MAX]`, um den Schwellenwert für Profilerstellungsvorgänge festzulegen.
- `profiler_sampling_rate` – Legen Sie auf einen Wert fest`[0.0-1.0]`, um den Prozentsatz der langsamen Operationen für das Profil festzulegen.

Weitere Informationen finden Sie unter Profilierung von Amazon DocumentDB-Vorgängen.

3. Profiler-Protokolle – Profiler-Protokolle nach Amazon exportieren CloudWatch
4. IAM-Rolle – Wählen Sie aus der Dropdown-Liste die Option RDS-Service-verknüpfte Rolle aus.

- Tags
  1. Tag hinzufügen – Geben Sie im Feld Schlüssel den Namen für das Tag für Ihren Cluster ein. Geben Sie optional im Feld Value (Wert) den Tag-Wert ein. Tags werden mit AWS Identity and Access Management (IAM)-Richtlinien verwendet, um den Zugriff auf Amazon DocumentDB-Ressourcen zu verwalten und zu steuern, welche Aktionen auf die Ressourcen angewendet werden können.
- Löschschutz
  1. Aktivieren des Löschschutzes – Schutz den Cluster davor, versehentlich gelöscht zu werden. Wenn diese Option aktiviert ist, können Sie den Cluster nicht löschen.

8. Um den Cluster wiederherzustellen, wählen Sie Create cluster (Cluster erstellen). Alternativ können Sie Cancel (Abbrechen) auswählen, um den Vorgang abzubrechen.

Using the AWS CLI

Um einen Cluster zu einem bestimmten Zeitpunkt mit Hilfe der Aufbewahrungsdauer des Snapshots wiederherzustellen, verwenden Sie die Operation `restore-db-cluster-to-point-in-time` mit den folgenden Parametern.

- **`--db-cluster-identifier`**– Erforderlich. Der Name des neuen Clusters, der erstellt werden soll. Dieser Cluster darf vor der Operation nicht vorhanden sein. Der Parameterwert muss den folgenden Einschränkungen entsprechen.

  Einschränkungen bei der Benennung von Clustern:

  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.

  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
- **`--restore-to-time`** – Das UTC-Datum und die UTC-Uhrzeit für die Wiederherstellung des Clusters. Beispiel: `2018-06-07T23:45:00Z`

  Zeiteinschränkungen:
  - Muss vor dem letzten wiederherstellbaren Zeitpunkt für den Cluster liegen.
  - Muss angegeben werden, wenn der Parameter `--use-latest-restorable-time` nicht angegeben ist.
  - Kann nicht angegeben werden, wenn der Parameter `--use-latest-restorable-time` auf `true` festgelegt ist.
  - Kann nicht angegeben werden, wenn der Parameterwert `--restore-type copy-on-write` ist.
- **`--source-db-cluster-identifier`** – Der Name des Quell-Clusters, aus dem wiederhergestellt werden soll. Dieser Cluster muss vorhanden und verfügbar sein.
- **`--use-latest-restorable-time`** oder **`--no-use-latest-restorable-time`** – Gibt an, ob die Wiederherstellung auf den letzten wiederherstellbaren Sicherungszeitpunkt erfolgen soll. Darf nicht angegeben werden, wenn der Parameter `--restore-to-time` angegeben ist.
- **`--storage-type standard | iopt1`** – Optional. Standard: `standard`.

Der AWS CLI Vorgang stellt `restore-db-cluster-to-point-in-time` nur den Cluster wieder her, nicht die Instances für diesen Cluster. Sie müssen die Operation `create-db-instance` aufrufen, um Instances für den wiederhergestellten Cluster zu erstellen, wobei Sie in `--db-cluster-identifier` die ID des wiederhergestellten Clusters angeben. Sie können Instances erst erstellen, nachdem die Operation `restore-db-cluster-to-point-in-time` abgeschlossen wurde und wenn der wiederhergestellte Cluster verfügbar ist.

Example

Das folgende Beispiel erstellt `sample-cluster-restored` aus Snapshot `sample-cluster-snapshot` für den letzten wiederherstellbaren Zeitpunkt.

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-restored \
    --source-db-cluster-identifier sample-cluster-snapshot \
    --use-latest-restorable-time
```

Für Windows:

```
aws docdb restore-db-cluster-to-point-in-time ^
    --db-cluster-identifier sample-cluster-restored ^
    --source-db-cluster-identifier sample-cluster-snapshot ^
    --use-latest-restorable-time
```

Example

Das folgende Beispiel erstellt `sample-cluster-restored` aus Snapshot `sample-cluster-snapshot` für 03:15 Uhr am 11. Dezember 2018 (UTC) (liegt innerhalb des Aufbewahrungszeitraums für Backups von `sample-cluster`).

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-restore \
    --source-db-cluster-identifier sample-cluster \
    --restore-to-time 2020-05-12T03:15:00Z
```

Für Windows:

```
aws docdb restore-db-cluster-to-point-in-time ^
    --db-cluster-identifier sample-cluster-restore ^
    --source-db-cluster-identifier sample-cluster ^
    --restore-to-time 2020-05-12T03:15:00Z
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-west-2b",
            "us-west-2a"
```

```
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster-restored",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-restored.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-restored.node.us-
east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "Port": 27017,
        "MasterUsername": "master-user",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abc0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID^>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-
restored",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-04-24T20:14:36.713Z",
        "DeletionProtection": false
    }
}
```

Löschen eines Cluster-Snapshots

Ein manueller Snapshot ist ein vollständiges Backup, das nur gelöscht wird, wenn Sie ihn manuell mit der AWS Management Console oder löschen AWS CLI. Sie können einen automatischen Snapshot nicht manuell löschen, da automatische Snapshots nur gelöscht werden, wenn die Aufbewahrungsfrist des Snapshots abläuft oder Sie den Cluster des Snapshots löschen.

Using the AWS Management Console

Führen Sie die folgenden Schritte aus AWS Management Console, um einen manuellen Cluster-Snapshot mit der zu löschen.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie in der Liste der Snapshots die Schaltfläche links neben dem Snapshot, den Sie löschen möchten. Der Typ des Snapshots muss manuell sein.

   1. Sie können überprüfen, ob der Typ des Snapshots manuell ist, indem Sie überprüfen, ob er als `manual` oder `automatic` in der Spalte Typ aufgeführt ist.
4. Wählen Sie im Menü Actions (Aktionen) die Option Delete (Löschen) aus. Wenn die Option Delete (Löschen) nicht verfügbar ist, haben Sie wahrscheinlich einen automatischen Snapshot auswählt.
5. Um den Snapshot zu löschen, wählen Sie auf der Bestätigungsseite des Löschvorgangs Delete (Löschen) aus. Um den Snapshot zu erhalten, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Ein manueller Amazon DocumentDB-Cluster-Snapshot ist ein vollständiges Backup, das Sie mit der manuell löschen können AWS CLI. Sie können einen automatischen Snapshot nicht manuell löschen.

Um einen manuellen Cluster-Snapshot mit der zu löschen AWS CLI, verwenden Sie die `-delete-db-cluster-snapshot`Operation mit den folgenden Parametern.

Parameter

- **`--db-cluster-snapshot-identifier`** – Erforderlich. Der Name des zu löschenden manuellen Snapshots.

Das folgende Beispiel löscht den Cluster-Snapshot `sample-cluster-snapshot`.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Für Windows:

```
aws docdb delete-db-cluster-snapshot ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Die Ausgabe dieses Vorgangs führt die Einzelheiten des von Ihnen gelöschten Cluster-Snapshots auf.

Verwaltung von Amazon DocumentDB-Ressourcen

In diesen Abschnitten werden die verschiedenen Komponenten und die damit verbundenen Aufgaben für die Verwaltung Ihrer Amazon DocumentDB-Implementierung (mit MongoDB-Kompatibilität) behandelt.

Themen

Amazon DocumentDB -Aufgaben

Dieser Abschnitt behandelt die operativen Aufgaben für Amazon DocumentDB -Cluster (mit MongoDB-Kompatibilität) und wie diese Aufgaben unter Verwendung der erledigt werden könnenAWS CLIaus.

Themen

- Erstellen eines Cluster-Snapshots
- Wiederherstellung aus einem Snapshot
- Entfernen einer Instance aus einem Cluster
- Löschen eines Clusters

Hinzufügen eines Replikats zu einem Amazon DocumentDB DocumentDB-Cluster

Nachdem Sie die primäre Instance für Ihr Amazon DocumentDB DocumentDB-Cluster erstellt haben, können Sie eine oder mehrere hinzufügenNachbildungenaus. Ein Replikat ist eine schreibgeschützte Instance, die zwei Zwecken dient:

- Skalierbarkeit— Wenn Sie über eine große Anzahl von Clients verfügen, die gleichzeitig auf zugreifen müssen, können Sie weitere Replicas für die Leseskalierung hinzufügen.
- Hohe Verfügbarkeit— Wenn die primäre Instance ausfällt, schaltet Amazon DocumentDB automatisch auf eine Replikat-Instance um und bestimmt sie als neue Primär-Instance. Wenn ein Replikat ausfällt, können andere Instances im Cluster nach wie vor für die Bearbeitung von Anfragen verwendet werden, bis der ausgefallene Knoten wiederhergestellt werden kann.

Jeder Amazon DocumentDB DocumentDB-Cluster kann bis zu 15 Replikate unterstützen.

> **Note**
>
> Für maximale Fehlertoleranz sollten Sie Replikate in separaten Availability Zones bereitstellen. So kann sichergestellt werden, dass Ihr Amazon DocumentDB DocumentDB-Cluster weiterhin ausgeführt werden kann, auch wenn die gesamte Availability Zone nicht mehr verfügbar sein sollte.

Im folgenden AWS CLI-Beispiel wird gezeigt, wie Sie ein neues Replikat hinzufügen. Der `--availability-zone`-Parameter platziert das Replikat in der angegebenen Availability Zone.

```
aws docdb create-db-instance \
    --db-instance-identifier sample-instance \
    --db-cluster-identifier sample-cluster \
    --engine docdb \
    --db-instance-class db.r5.large \
```

```
    --availability-zone us-east-1a
```

Beschreiben von Clustern und Instances

FolgendesAWS CLIBeispiel: listet alle Amazon DocumentDB DocumentDB-Cluster in einer Region auf. Für bestimmte Verwaltungsfunktionen, z. B. Cluster- und Instance-Lifecycle-Management, nutzt Amazon DocumentDB die betriebliche Technologie, die mit Amazon RDS gemeinsam genutzt wird. Die`filterName=engine,Values=docdb`Der Filterparameter gibt nur Amazon DocumentDB DocumentDB-Cluster zurück.

Weitere Informationen zum Beschreiben und Ändern von Clustern finden Sie im Amazon DocumentDB-Cluster-Lebenszyklus.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusters": [
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-1",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-2",
            "DBClusterParameterGroup": "sample-parameter-group",
```

```
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-3",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        }
    ]
}
```

FolgendesAWS CLIIn beispiel werden die Instances in einem Amazon DocumentDB DocumentDB-Cluster aufgelistet. Weitere Informationen zum Beschreiben und Ändern von Clustern finden Sie im Amazon DocumentDB-Instance-Lebenszyklus.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Die Ausgabe sieht wie unten aus. Diese Ausgabe enthält zwei Instances. Die Primär-Instance ist `sample-instance-1` (`"IsClusterWriter": true`). Es gibt auch eine Replikat-Instance, nämlich `sample-instance2` (`"IsClusterWriter: false"`).

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
```

```
                "DBInstanceIdentifier": "sample-cluster-2",
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ]
    ]
]
```

Erstellen eines Cluster-Snapshots

EIN-Cluster-Snapshotist eine vollständige Sicherung der Daten in Ihrem Amazon DocumentDB DocumentDB-Cluster. Wenn der Snapshot erstellt wird, liest Amazon DocumentDB Ihre Daten direkt aus dem Cluster-Volume. Aus diesem Grund können Sie einen Snapshot erstellen, auch wenn zum jeweiligen Zeitpunkt keine Instances in Ihrem Cluster ausgeführt werden. Die zum Erstellen eines Snapshots erforderliche Zeit hängt von der Größe Ihres Cluster-Volumes ab.

Amazon DocumentDB unterstützt automatische Backups, die täglich während des bevorzugten Backup-Fensters auftreten — ein Zeitraum von 30 Minuten während des Tages. Im folgenden AWS CLI-Beispiel wird veranschaulicht, wie Sie das Sicherungszeitfenster für Ihr Cluster anzeigen:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].PreferredBackupWindow'
```

Die Ausgabe zeigt das Sicherungszeitfenster (in UTC):

```
[
    "00:18-00:48"
]
```

Sie können das Sicherungszeitfenster bei der Erstellung Ihres Amazon DocumentDB DocumentDB-Clusters definieren. Sie können das Sicherungszeitfenster auch ändern, wie im folgenden Beispiel gezeigt wird. Wenn Sie kein Sicherungszeitfenster festlegen, weist Amazon DocumentDB Ihrem Cluster automatisch ein Fenster zu.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --preferred-backup-window "02:00-02:30"
```

Zusätzlich zu automatischen Sicherungen können Sie jederzeit manuell einen Cluster-Snapshot erstellen. Wenn Sie dies tun, müssen Sie das zu sichernde Cluster festlegen und dem Snapshot einen eindeutigen Namen geben, damit er später für Wiederherstellungszwecke verwendet werden kann.

Im folgenden AWS CLI-Beispiel wird gezeigt, wie Sie einen Snapshot Ihrer Daten erstellen.

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Wiederherstellung aus einem Snapshot

Sie können einen Cluster-Snapshot in einem neuen Amazon DocumentDB DocumentDB-Cluster wiederherstellen. Dazu geben Sie den Namen des Snapshots und den Namen eines neuen Clusters an. Sie können keine Wiederherstellung von einem Snapshot auf einem bestehenden -Cluster durchführen. Stattdessen erstellt Amazon DocumentDB bei der Wiederherstellung einen neuen -Cluster und füllt ihn dann mit Ihren Snapshot-Daten.

Das folgende Beispiel zeigt alle Snapshots für den Cluster `sample-cluster`.

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotType,Status]'
```

Die Ausgabe sieht ungefähr wie folgt aus. Ein manueller Snapshot ist ein Snapshot, den Sie manuell erstellt haben. Ein automatisierter Snapshot hingegen ist ein Snapshot.

```
[
        "sample-cluster-snapshot",
        "manual",
        "available"
    ],
    [
        "rds:sample-cluster",
        "automated",
        "available"
    ]
]
```

Im folgenden Beispiel wird gezeigt, wie Sie einen Amazon DocumentDB DocumentDB-Cluster aus einem Snapshot wiederherstellen.

```
aws docdb restore-db-cluster-from-snapshot \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster \
    --snapshot-identifier sample-cluster-snapshot
```

Dem neuen Cluster sind keine Instances zugeordnet. Wenn Sie mit dem Cluster interagieren möchten, müssen Sie ihm also eine Instance hinzufügen.

```
aws docdb create-db-instance \
    --db-instance-identifier new-sample-instance \
    --db-instance-class db.r5.large \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster
```

Sie können die folgenden AWS CLI-Operationen verwenden, um den Fortschritt der Cluster- und Instance-Erstellung zu überwachen. Wenn die Cluster- und Instance-Status verfügbar sind, können Sie eine Verbindung zum neuen Endpunkt des Clusters herstellen und auf Ihre Daten zugreifen.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier new-sample-cluster  \
    --query 'DBClusters[*].[Status,Endpoint]'
```

```
aws docdb describe-db-instances \
    --db-instance-identifier new-sample-instance \
    --query 'DBInstances[*].[DBInstanceStatus]'
```

Entfernen einer Instance aus einem Cluster

Amazon DocumentDB speichert alle Daten im Cluster-Volume. Die Daten bleiben in diesem Cluster-Volume erhalten. Dies gilt auch dann, wenn Sie alle Instances von Ihrem Cluster entfernen. Wenn Sie erneut auf die Daten zugreifen müssen, können Sie jederzeit dem Cluster eine Instance hinzufügen und da weitermachen, wo Sie aufgehört haben.

Im folgenden Beispiel wird gezeigt, wie Sie eine Instance aus Ihrem Amazon DocumentDB DocumentDB-Cluster entfernen.

```
aws docdb delete-db-instance \
```

```
    --db-instance-identifier sample-instance
```

Löschen eines Clusters

Bevor Sie einen Amazon DocumentDB DocumentDB-Cluster löschen können, müssen Sie zunächst alle seine Instances entfernen. Im folgenden AWS CLI-Beispiel werden Informationen über die Instances in einem Cluster zurückgegeben. Wenn diese Operation Instance-Kennungen zurückgibt, müssen Sie die einzelnen Instances löschen. Weitere Informationen finden Sie unter Entfernen einer Instance aus einem Cluster.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterMembers[*].DBInstanceIdentifier'
```

Wenn keine Instances mehr vorhanden sind, können Sie das Cluster löschen. Sie müssen dann eine der folgenden Optionen wählen:

- Erstellen Sie einen endgültigen Snapshot— Erfassen Sie alle Clusterdaten in einem Snapshot, damit Sie später eine neue Instanz mit diesen Daten neu erstellen können. Das Verfahren wird im folgenden Beispiel beschrieben:

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --final-db-snapshot-identifier sample-cluster-snapshot
  ```

- Überspringen Sie den letzten Snapshot- Verwerfen Sie alle Clusterdaten dauerhaft. Diese Aktion ist unwiderruflich. Das Verfahren wird im folgenden Beispiel beschrieben:

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --skip-final-snapshot
  ```

Übersicht über globale Amazon DocumentDB-Cluster

Was ist ein globaler Cluster?

Ein globaler Cluster besteht aus einer primären Region und bis zu fünf schreibgeschützten sekundären Regionen. Sie geben Schreibvorgänge direkt an den primären Cluster in der primären

Region aus und Amazon DocumentDB repliziert die Daten mithilfe einer dedizierten Infrastruktur automatisch in die sekundären Regionen. Die Latenz liegt in der Regel unter einer Sekunde.

Wie sind globale Cluster nützlich?

- Wiederherstellung nach regionsweiten Ausfällen – Im Falle eines regionsweiten Ausfalls können Sie innerhalb von Minuten einen der sekundären Cluster zu einem primären Cluster hochstufen, wobei das typische Recovery Time Objective (RTO) unter einer Minute liegt. Das Recovery Point Objective (RPO) wird in der Regel in Sekunden gemessen, dies hängt jedoch von der Verzögerung im gesamten Netzwerk zum Zeitpunkt des Ausfalls ab.
- Globale Lesevorgänge mit lokaler Latenz – Wenn Sie Niederlassungen auf der ganzen Welt haben, können Sie einen globalen Cluster verwenden, um Ihre wichtigsten Datenquellen in der primären Region auf dem neuesten Stand zu halten. Niederlassungen in Ihren anderen Regionen können mit lokaler Latenz auf die Informationen in ihrer eigenen Region zugreifen.
- Skalierbare sekundäre Cluster – Sie können Ihre sekundären Cluster skalieren, indem Sie einer sekundären Region weitere schreibgeschützte Instances hinzufügen. Der sekundäre Cluster ist schreibgeschützt und kann daher bis zu 16 schreibgeschützte Replikat-Instances anstelle des üblichen Limits von 15 für einen einzelnen Cluster unterstützen.
- Schnelle Replikation von primären zu sekundären Clustern – Die von einem globalen Cluster durchgeführte Replikation hat nur geringe Auswirkungen auf die Leistung des primären Datenbank-Clusters. Die Ressourcen der DB-Instances werden ausschließlich für Lese- und Schreib-Workloads von Anwendungen genutzt.

Was sind die aktuellen Einschränkungen globaler Cluster?

- Globale Cluster werden auf Amazon DocumentDB v3.6 nicht unterstützt.
- Globale Cluster werden auf den Instance-Typen t3, t4g und r4 nicht unterstützt.
- Globale Cluster sind in den folgenden Regionen nicht verfügbar: Südamerika (São Paulo), Europa (Mailand), China (Peking) und China (Ningxia).
- Im Falle eines regionalen Failovers müssen Sie einen sekundären Cluster manuell zum primären Cluster hochstufen und Ihre Anwendung so ändern, dass sie auf den neuen primären Cluster verweist.
- Nur der primäre Cluster führt Schreibvorgänge aus. Clients, die Schreibvorgänge ausführen, stellen eine Verbindung zum Cluster-Endpunkt des primären Clusters her.
- Sie können maximal fünf sekundäre Regionen und eine primäre Region für Ihren Cluster haben.

- Ein sekundärer Cluster kann nicht gestoppt werden. Ein primärer Cluster kann nicht gestoppt werden, wenn ihm sekundäre Cluster zugeordnet sind. Nur ein regionaler Cluster ohne sekundäre Cluster kann gestoppt werden.
- Dem sekundären Cluster angefügte Replikate können unter bestimmten Umständen neu gestartet werden. Wenn die Instance der primären Region neu gestartet wird oder ein Failover durchführt, werden Replikate in der sekundären Region ebenfalls neu gestartet. Der Cluster ist dann nicht verfügbar, bis alle Replikate wieder mit der Writer-Instance des Primärdatenbank-Clusters synchronisiert sind. Dieses Verhalten wird erwartet. Stellen Sie sicher, dass Sie die Auswirkungen auf Ihren globalen Cluster verstehen, bevor Sie Änderungen an Ihrem primären Cluster vornehmen.
- Sie können Änderungsstreams nicht auf sekundären Clustern verwenden.

Themen

Schnellstart: Globale Cluster

Themen

Konfiguration

Der globale Amazon DocumentDB-Cluster umfasst mindestens zwei AWS-Regionen. Die primäre Region unterstützt einen Cluster mit einer primären (Writer-)Instance und bis zu fünfzehn Replikat-Instances, während eine sekundäre Region einen schreibgeschützten Cluster ausführt, der vollständig aus bis zu sechszehn Replikat-Instances besteht. Ein globaler Cluster kann bis zu

fünf sekundäre Regionen haben. Die Tabelle listet die maximal zulässigen Cluster, Instances und Replikate in einem globalen Cluster auf.

Beschreibung	Primär AWS-Region	Sekundär AWS-Region
Cluster	1	5 (maximal)
Writer-Inst	1	0
Schreibgeschützte Instances (Amazon DocumentDB-Replikate) pro Cluster	15 (Max.)	16 (Total)
Schreibgeschützte Instances (maximal zulässig, bei tatsächlicher Anzahl von sekundären Regionen)	15 - s	s = Gesamtzahl der sekundäre n AWS-Regionen

Die Cluster haben die folgenden spezifischen Anforderungen:

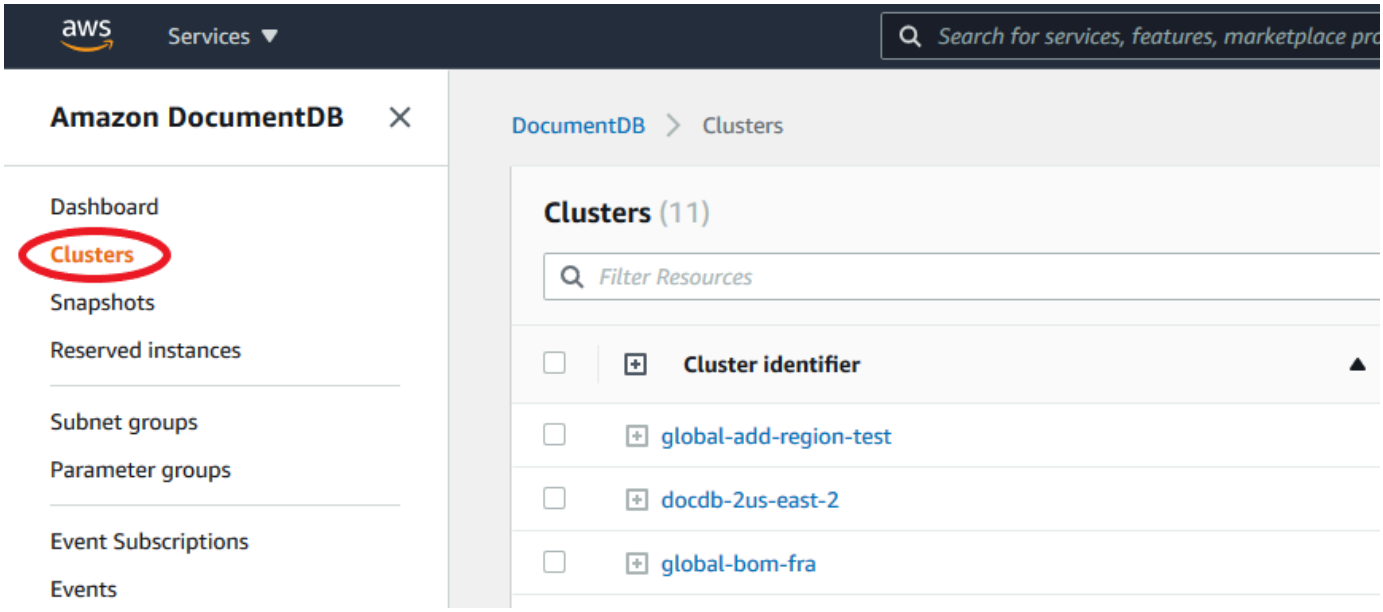
- Anforderungen an Datenbank-Instance-Klassen – Sie können nur die `db.r6` Instance-Klassen `db.r5` und verwenden.
- -AWS-RegionAnforderungen – Der primäre Cluster muss sich in einer Region befinden und mindestens ein sekundärer Cluster muss sich in einer anderen Region desselben Kontos befinden. Sie können bis zu fünf sekundäre (schreibgeschützte) Cluster erstellen, die sich jeweils in einer anderen Region befinden müssen. Mit anderen Worten, es können sich keine zwei Cluster in derselben Region befinden.
- Benennungsanforderungen – Die Namen, die Sie für jeden Ihrer Cluster auswählen, müssen in allen Regionen eindeutig sein. Sie können nicht denselben Namen für verschiedene Cluster verwenden, obwohl sie sich in verschiedenen Regionen befinden.

Erstellen eines globalen Amazon DocumentDB-Clusters

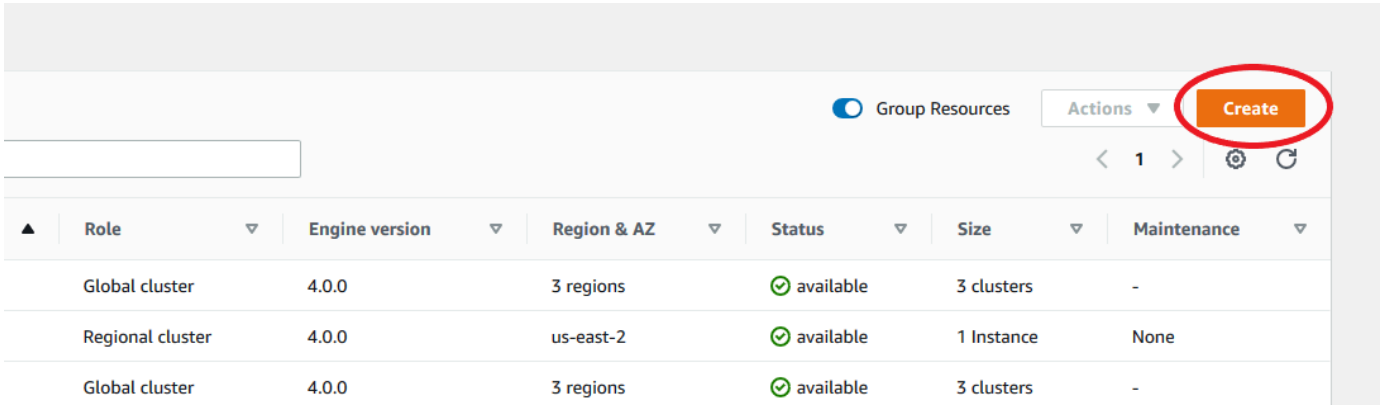
Sind Sie bereit, Ihren ersten globalen Cluster zu erstellen? In diesem Abschnitt wird erläutert, wie Sie einen völlig neuen globalen Cluster mit neuen Datenbank-Clustern und Instances erstellen, indem Sie entweder die AWS Management Console oder die AWS CLI mit den folgenden Anweisungen verwenden.

Verwenden des AWS Management Console

1. Navigieren AWS Management ConsoleSie in der zu Amazon DocumentDB .
2. Wenn Sie zur Amazon DocumentDB-Konsole gelangen, wählen Sie Cluster aus.

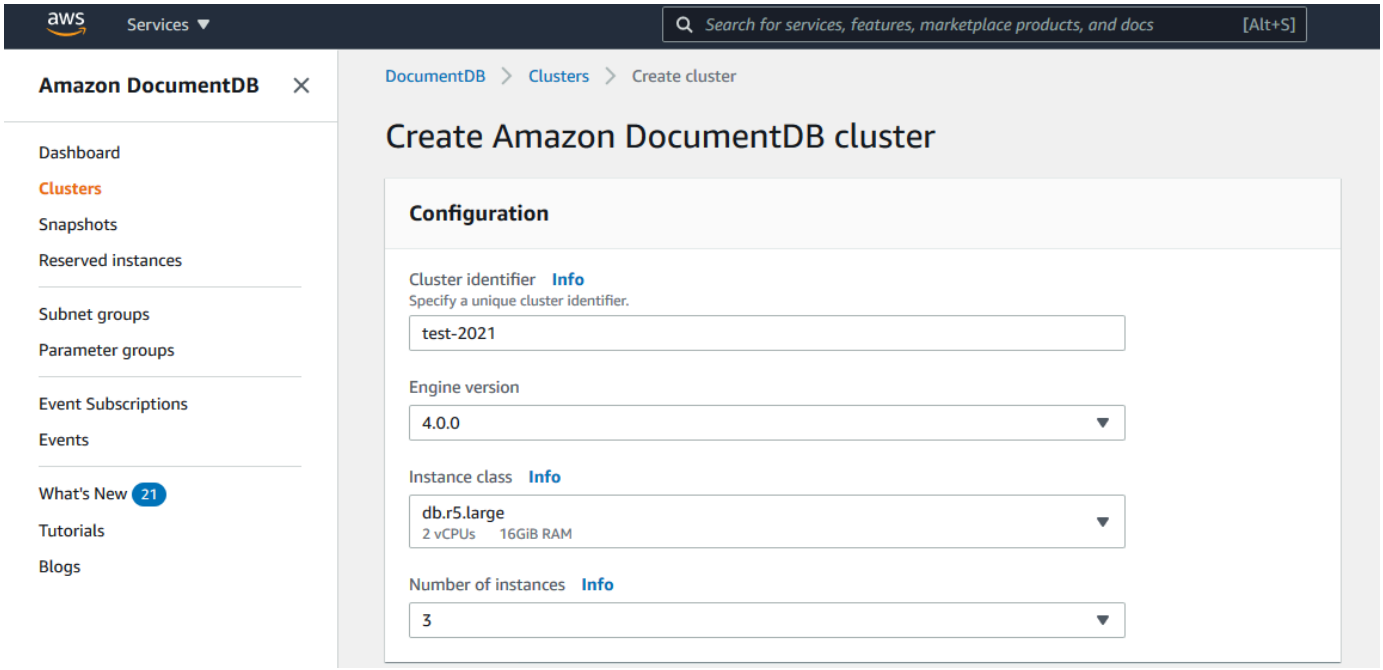


3. Wählen Sie Erstellen.

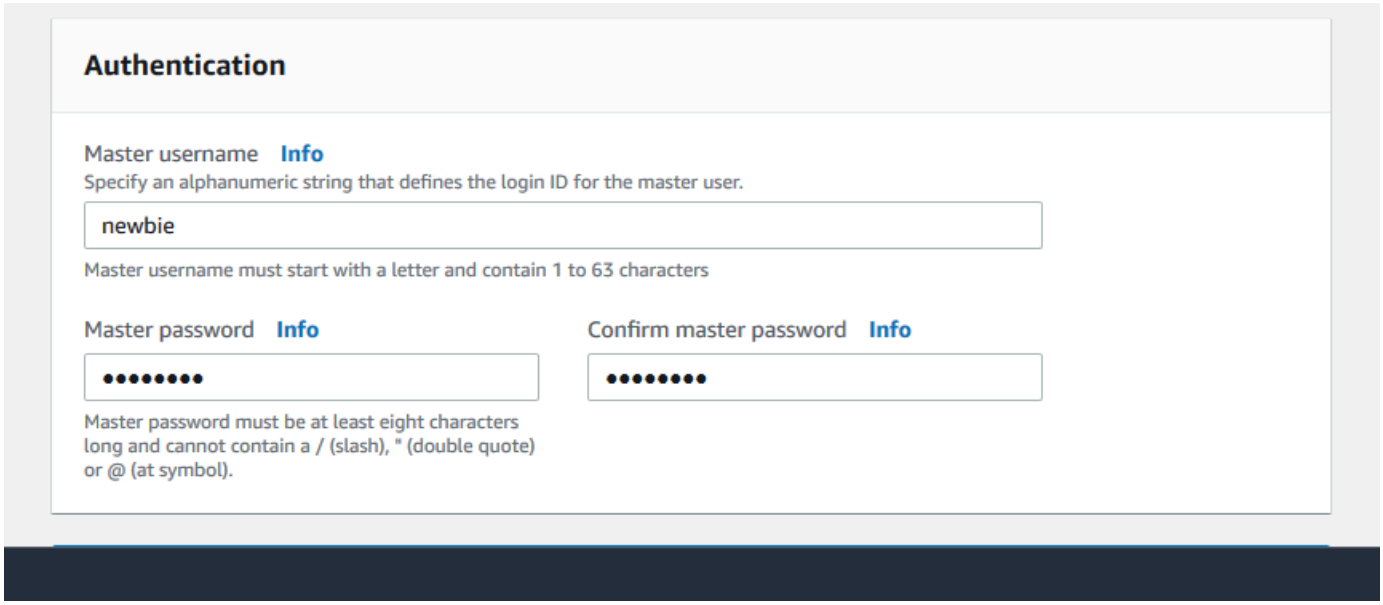


4. Füllen Sie den Abschnitt Konfiguration des Formulars Amazon DocumentDB-Cluster erstellen entsprechend aus:

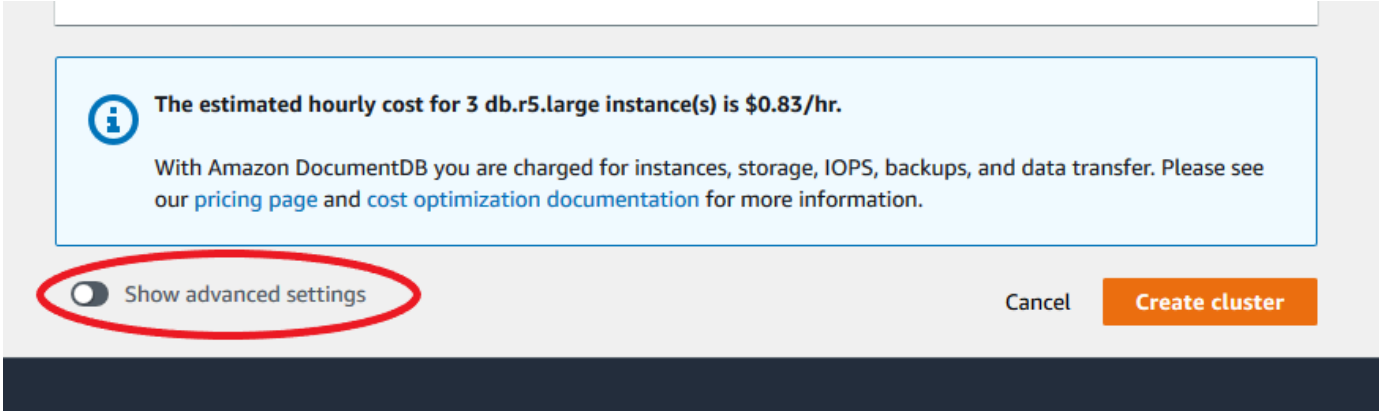
   - Cluster-ID: Sie können entweder eine eindeutige ID für diese Instance eingeben oder Amazon DocumentDB erlauben, die Instance-ID basierend auf der Cluster-ID bereitzustellen.
   - Engine-Version: Wählen Sie 4.0.0
   - Instance-Klasse: Wählen Sie db.r5.large
   - Anzahl der Instances: Wählen Sie 3 aus.



5. Geben Sie im Abschnitt Authentifizierung einen Hauptbenutzernamen und ein Hauptpasswort ein.

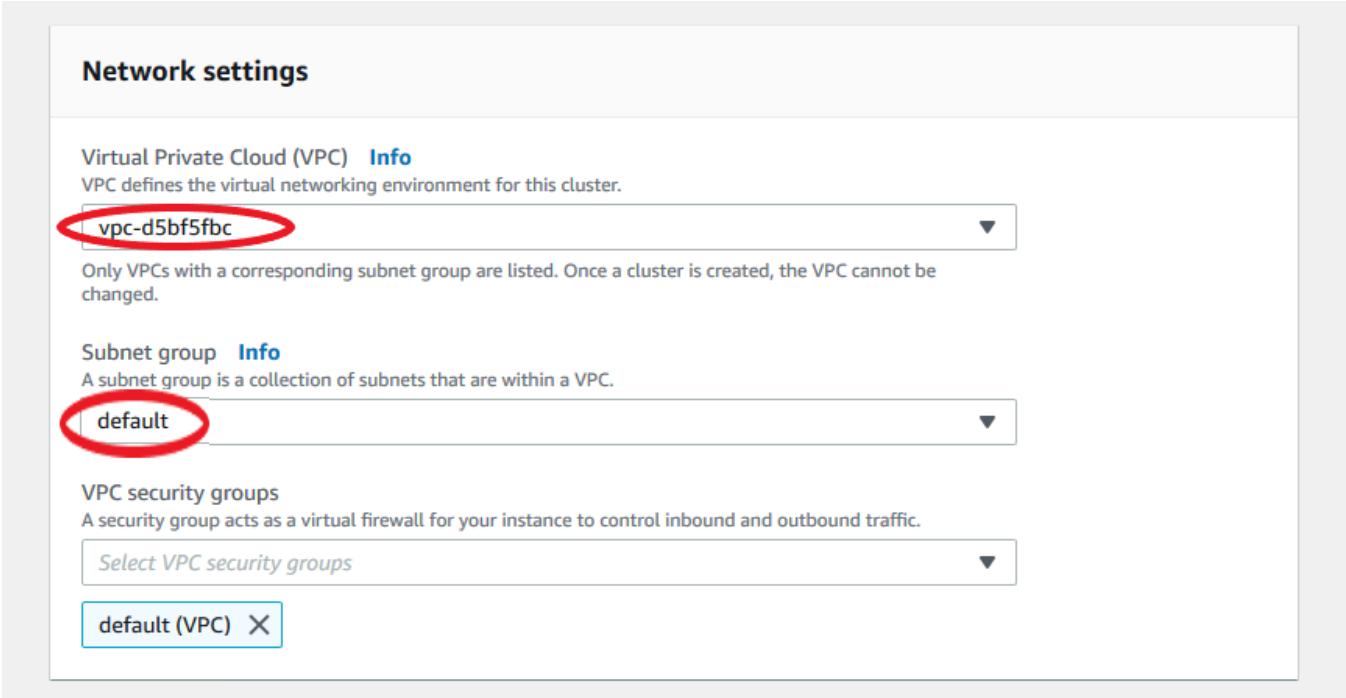


6. Wählen Sie Show Advanced Settings aus.

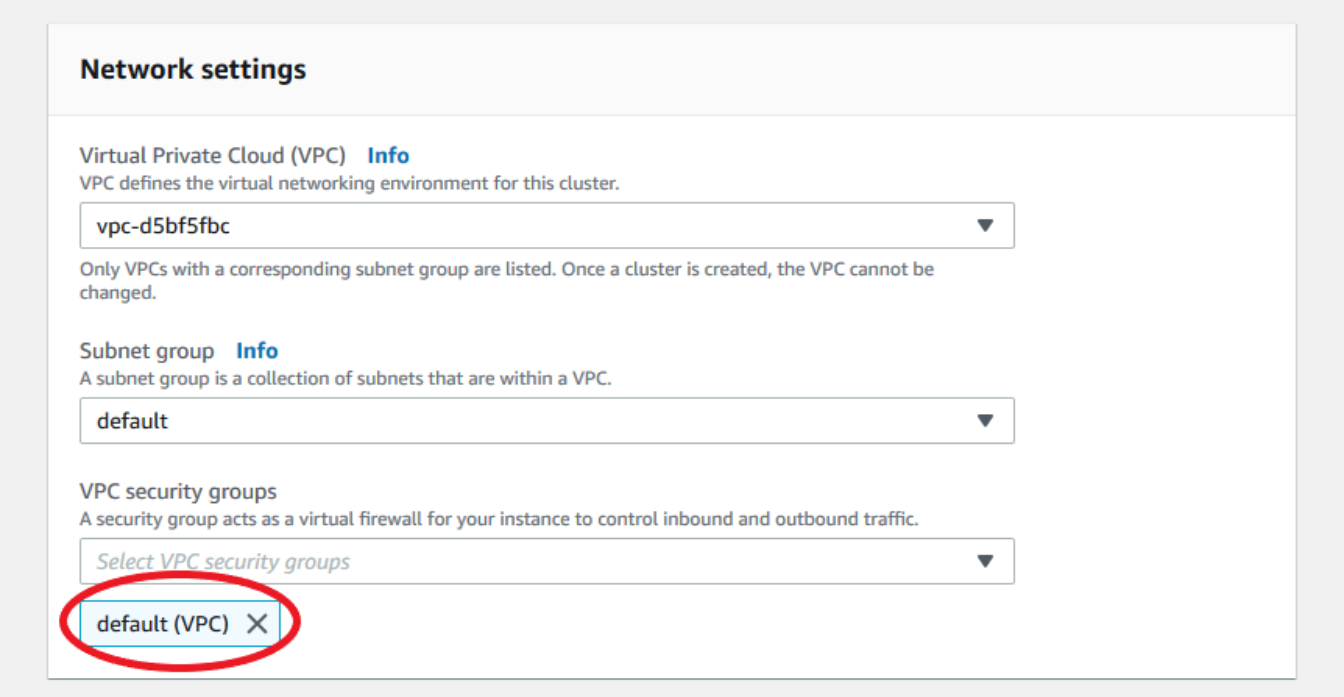


7. Im Abschnitt Netzwerkeinstellungen:

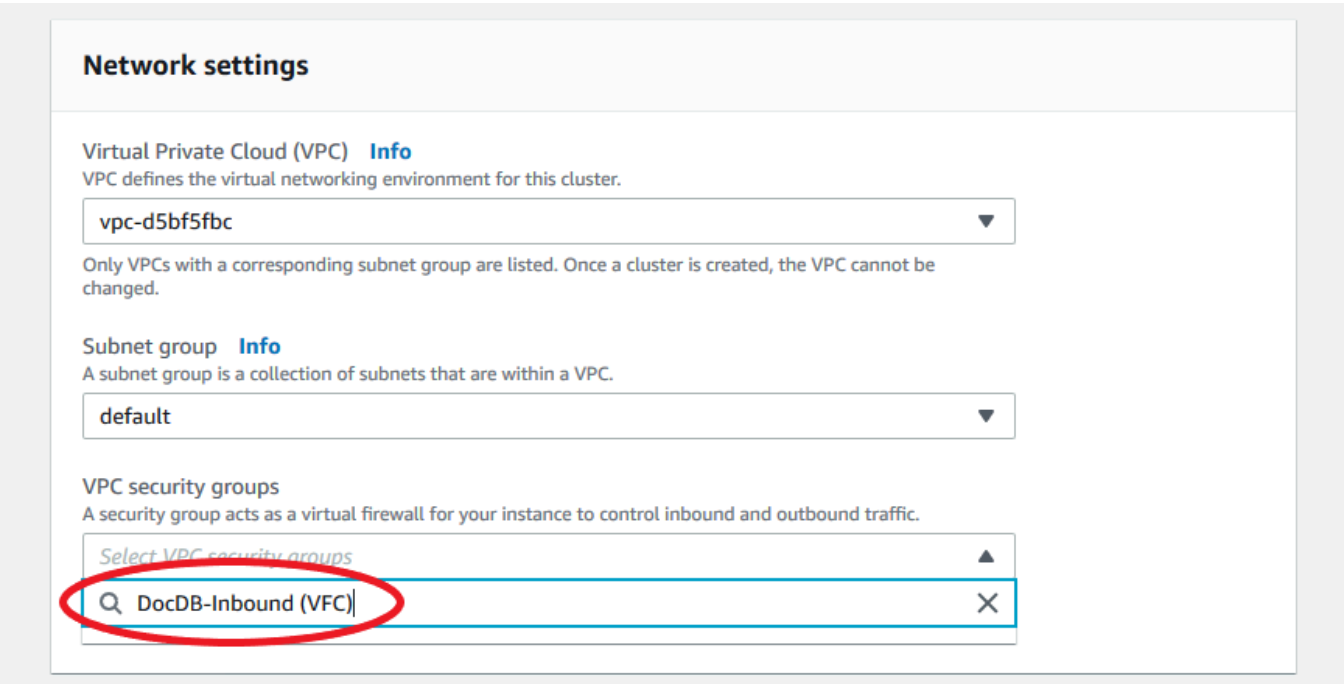
   - Behalten Sie die Standardoptionen für Virtual Private Cloud und Subnetzgruppe bei.



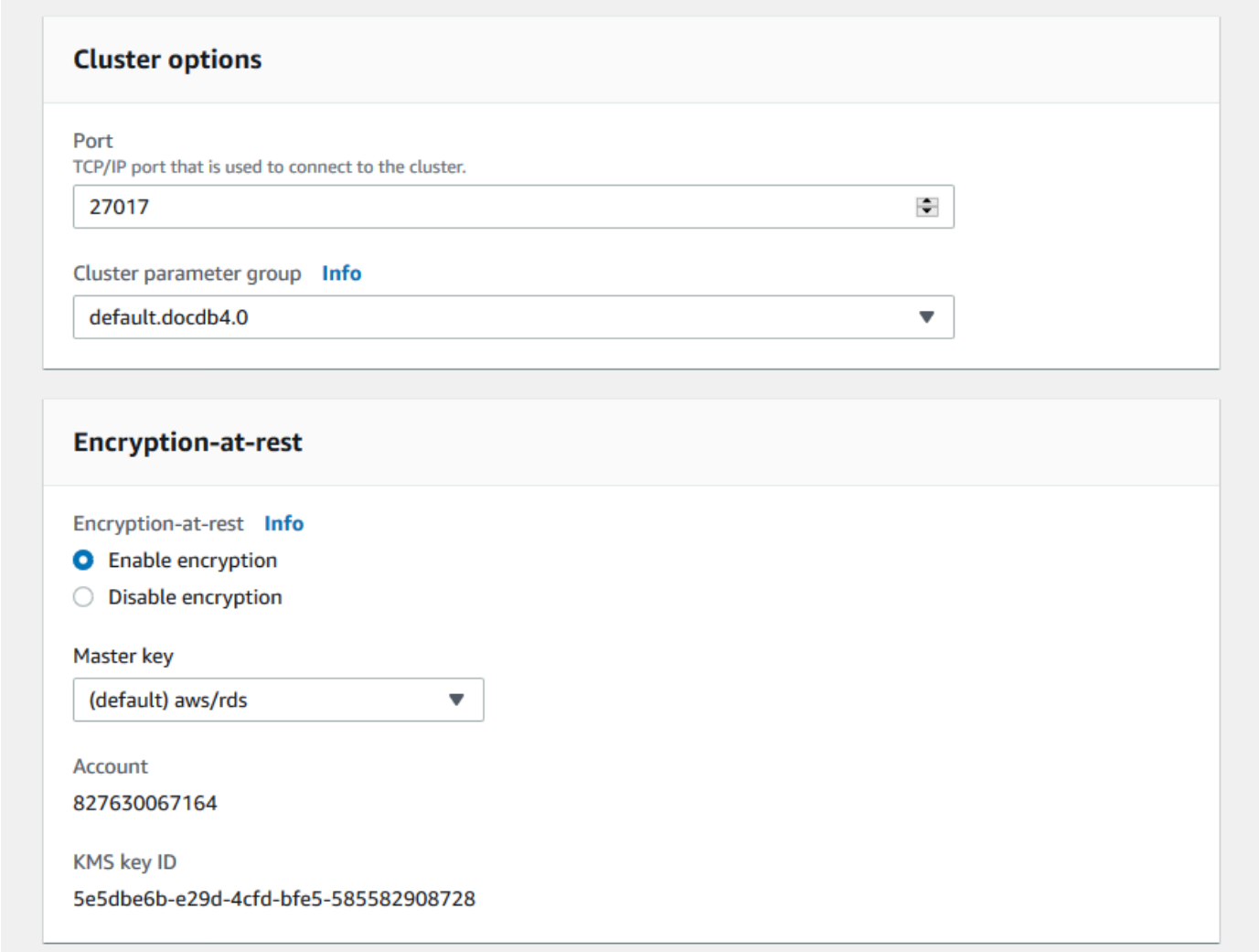
   - Für VPC-Sicherheitsgruppen sollte die Standard-VPC bereits hinzugefügt werden.



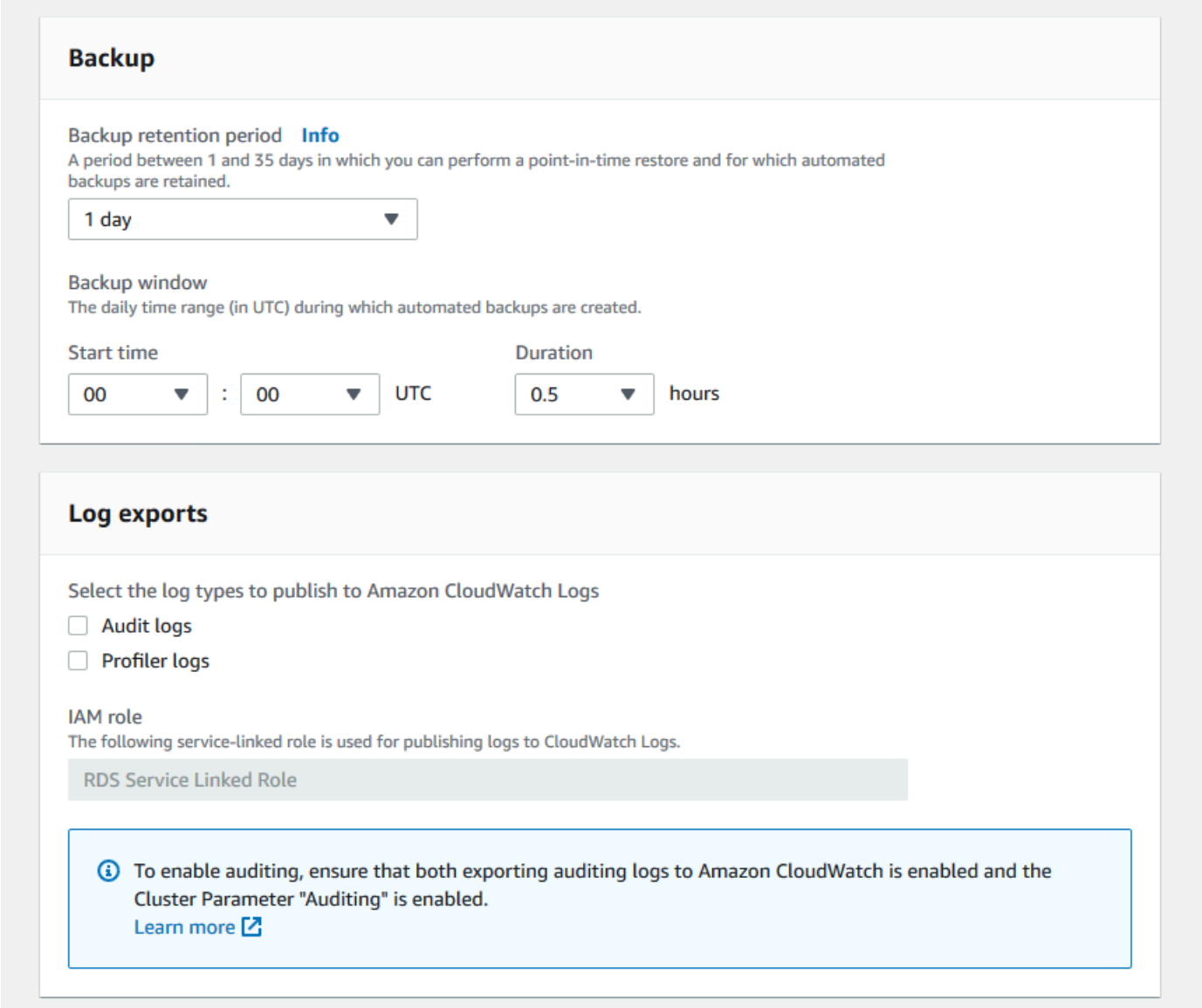
- Geben Sie DocDB in das Feld VPC-Sicherheitsgruppen ein und wählen Sie DocDB -Inbound (VPC) aus.



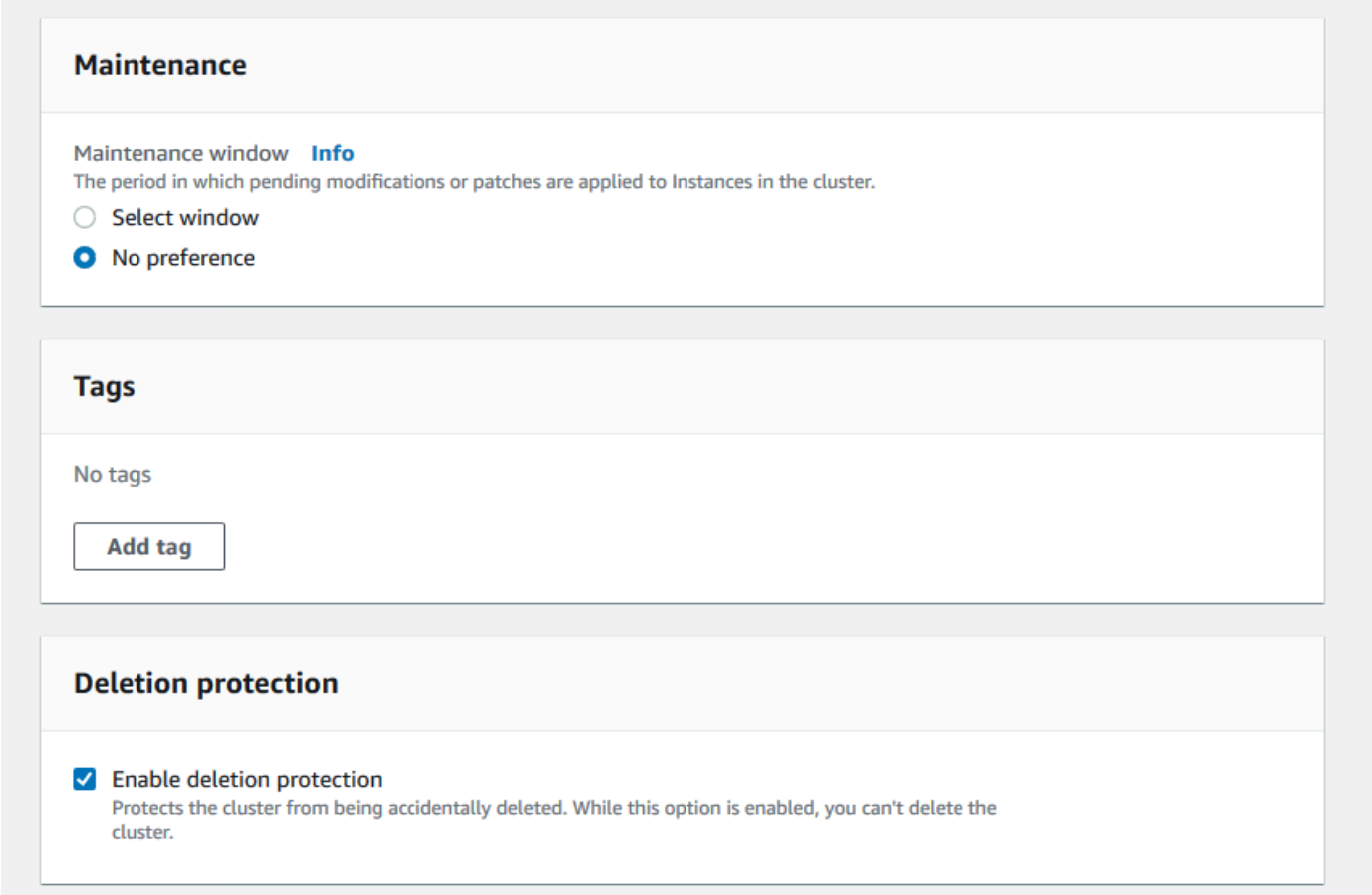
8. Behalten Sie für Cluster-Optionen und Encryption-at-rest die Standardauswahl bei.



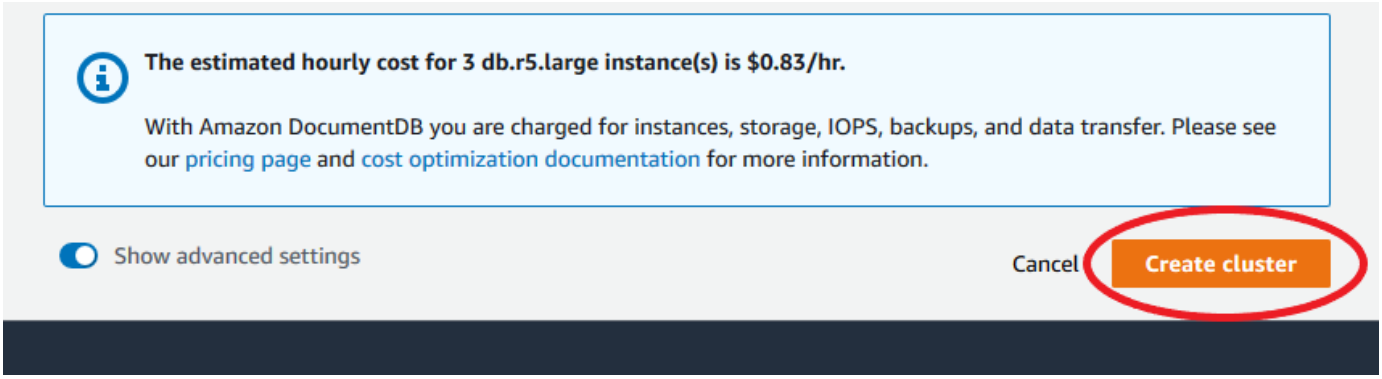
9. Behalten Sie für Backup -und Protokollexporte die Standardauswahl bei.



10. Behalten Sie für Wartung , Tags und Löschschutz die Standardauswahl bei.



11. Klicken Sie jetzt auf die Schaltfläche mit der Bezeichnung Erstellen.



Verwenden des AWS CLI

Um einen regionalen Amazon DocumentDB-Cluster zu erstellen, rufen Sie die `create-db-cluster` aufAWS CLI. Der folgende AWS CLI Befehl erstellt einen Amazon DocumentDB-Cluster mit dem Namen `global-cluster-id`. Weitere Informationen zum Löschschutz finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Außerdem `--engine-version` ist ein optionaler Parameter, der standardmäßig die neueste Engine-Hauptversion verwendet. Die aktuelle Engine-Hauptversion ist `4.0.0`. Wenn neue Engine-Hauptversionen veröffentlicht werden, `--engine-version` wird die Standard-Engine-Version für aktualisiert, sodass sie der letzten Engine-Hauptversion entspricht. Daher empfehlen wir, für Produktions-Workloads und insbesondere für Workloads, die von Skripten, Automatisierungen oder AWS CloudFormation Vorlagen abhängig sind, die explizit `--engine-version` für die gewünschte Hauptversion anzugeben.

Wenn `db-subnet-group-name` oder nicht angegeben `vpc-security-group-id` ist, verwendet Amazon DocumentDB die Standardsubnetzgruppe und die Amazon-VPC-Sicherheitsgruppe für die angegebene Region.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --global-cluster-identifier global-cluster-id \
      --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

Für Windows:

```
aws docdb create-db-cluster ^
      --global-cluster-identifier global-cluster-id ^
      --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "4.0.0",
```

```
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:cluster-id",
        "DBClusterIdentifier": "cluster-id",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "cluster-id.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "cluster-id.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

Die Erstellung des Clusters dauert mehrere Minuten. Sie können die AWS Management Console oder AWS CLI verwenden, um den Status Ihres Clusters zu überwachen. Weitere Informationen finden Sie unter Überwachung des Status eines Amazon DocumentDB-Clusters.

> **Important**
>
> Wenn Sie die verwendenAWS CLI, um einen regionalen Amazon DocumentDB-Cluster zu erstellen, werden keine Instances erstellt. Daher müssen Sie explizit eine primäre Instance und alle benötigten Replikat-Instances anlegen. Sie können entweder die Konsole oder die AWS CLI verwenden, um die Instances zu erstellen. Weitere Informationen finden Sie unter

Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster und CreateDBCluster in der Amazon DocumentDB-API-Referenz.

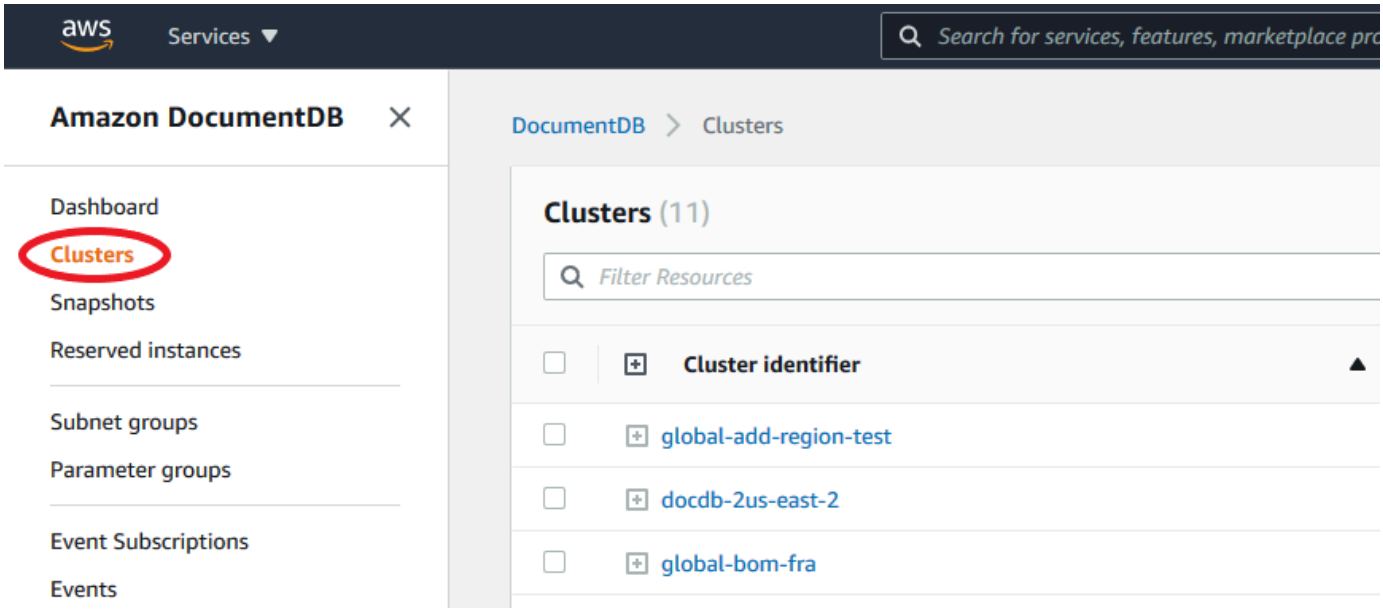
Sobald Ihr regionaler Cluster verfügbar ist, können Sie einen sekundären Cluster in einer anderen Region mit den folgenden Anweisungen hinzufügen: Hinzufügen eines AWS-Region zu einem globalen Amazon DocumentDB-Cluster. Wenn Sie eine Region hinzufügen, wird Ihr regionaler Cluster Ihr primärer Cluster und Sie haben einen neuen sekundären Cluster in der ausgewählten Region.

Hinzufügen eines AWS-Region zu einem globalen Amazon DocumentDB-Cluster

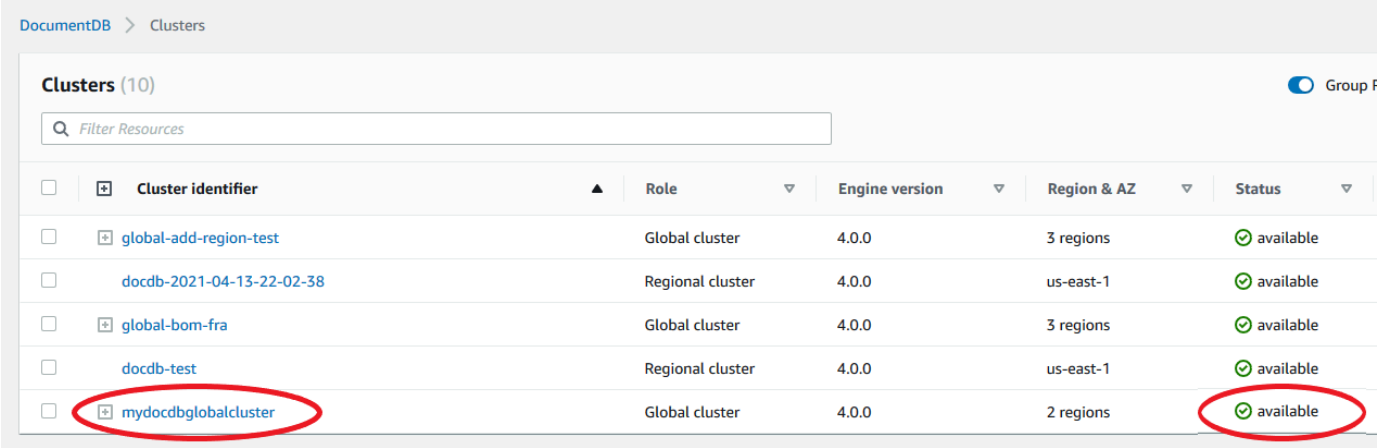
Ein globaler Cluster benötigt mindestens einen sekundären Cluster in einer anderen Region als der primäre Cluster, und Sie können bis zu fünf sekundäre Cluster hinzufügen. Beachten Sie, dass Sie für jeden sekundären Cluster, den Sie hinzufügen, die Anzahl der im primären Cluster zulässigen Replikate um eins reduzieren müssen. Wenn Ihr globaler Cluster beispielsweise fünf sekundäre Regionen hat, kann Ihr primärer Cluster nur zehn (statt fünfzehn) Replikate haben. Weitere Informationen finden Sie unter Konfigurationsanforderungen eines globalen Amazon DocumentDB-Clusters.

Verwenden des AWS Management Console

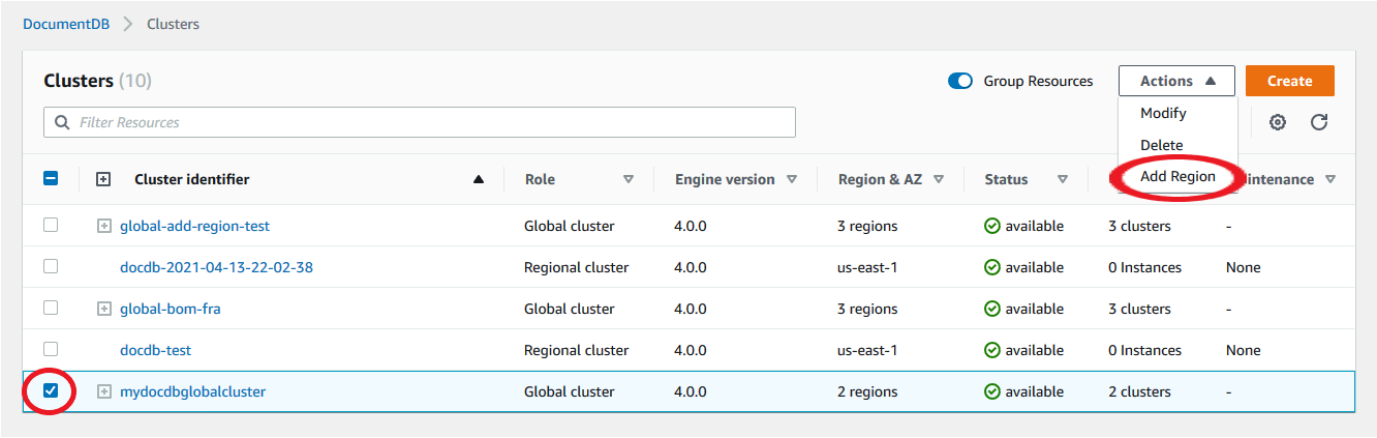
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.



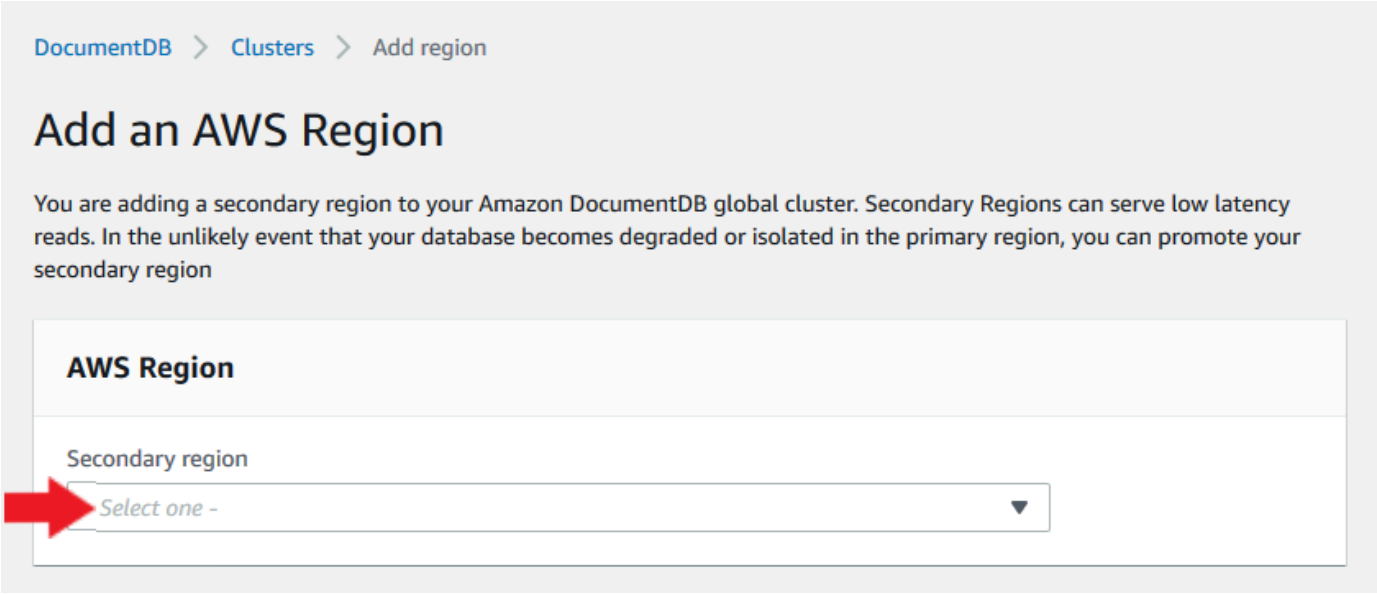
3. Wählen Sie den Cluster aus, dem Sie einen sekundären Cluster hinzufügen möchten. Stellen Sie sicher, dass der Cluster ist`Available`.



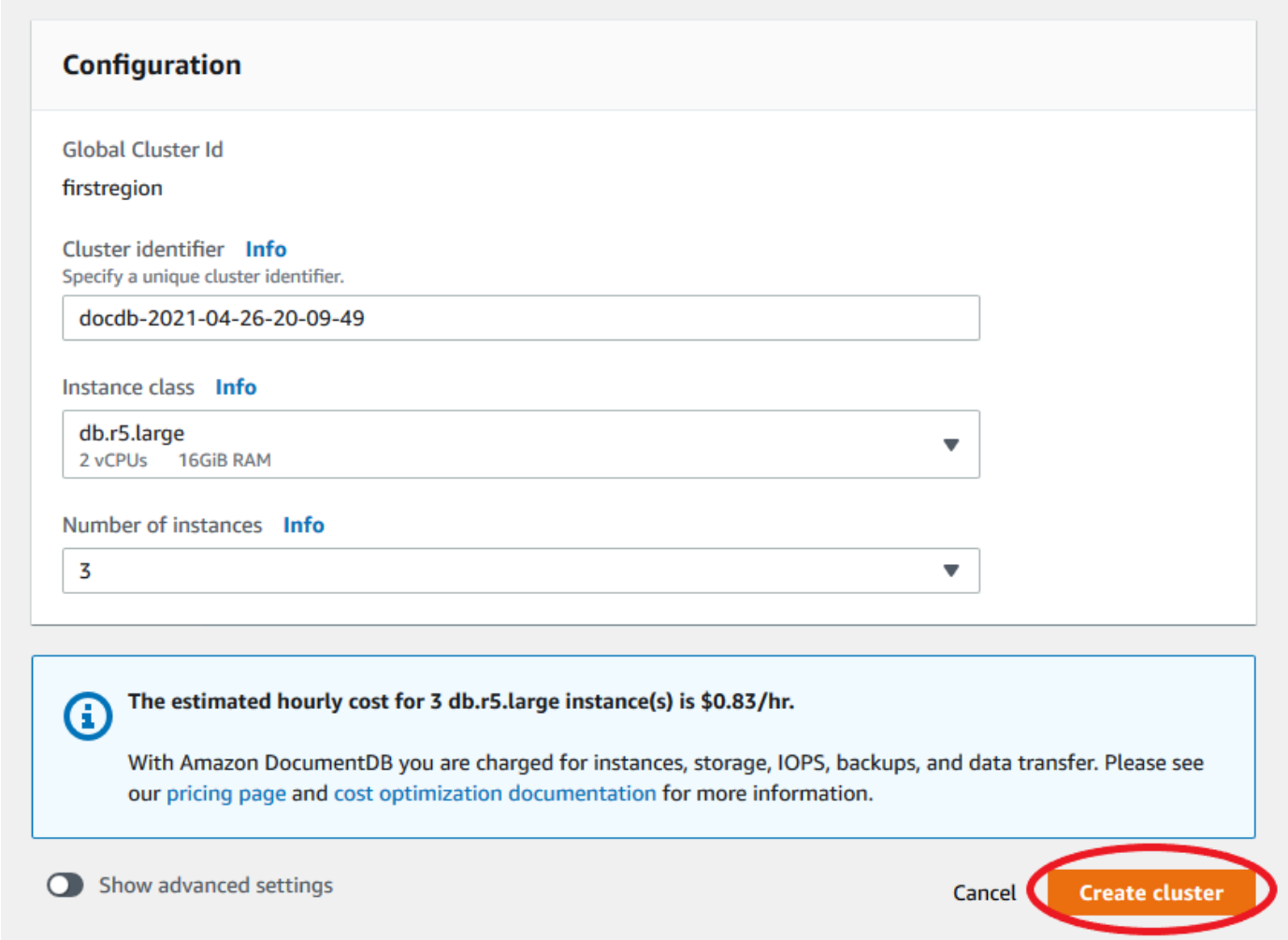
4. Wählen Sie das Dropdown-Menü für Aktionen und dann Region hinzufügen aus.



5. Wählen Sie auf der Seite Region hinzufügen die sekundäre Region aus. Beachten Sie, dass Sie keine Region auswählen können, die bereits über einen sekundären Cluster für denselben globalen Cluster verfügt. Außerdem kann es sich nicht um dieselbe Region wie der primäre Cluster handeln. Wenn Sie die erste Region hinzufügen, müssen Sie auch eine globale Cluster-ID Ihrer Wahl angeben.



6. Füllen Sie die verbleibenden Felder für den sekundären Cluster in der neuen Region aus und wählen Sie dann Cluster erstellen aus. Nachdem Sie die Region hinzugefügt haben, können Sie sie in der Liste der Cluster in der sehenAWS Management Console.



Verwenden des AWS CLI

- Verwenden Sie den `create-db-cluster` CLI-Befehl mit dem Namen Ihres globalen (`--global-cluster-identifier`) Clusters. Für andere Parameter, führen Sie die folgenden Schritte aus:
  - `--region`Wählen Sie für eine andere AWS-Region als die Ihrer primären Region aus.
  - Wählen Sie bestimmte Werte für die Parameter `--engine-version` und `--engine` aus.
  - Geben Sie für einen verschlüsselten Cluster Ihre primäre AWS-Region als `--source-region` für die Verschlüsselung an.

Im folgenden Beispiel wird ein neuer Amazon DocumentDB-Cluster erstellt und als schreibgeschützter sekundärer Cluster an den globalen Cluster angehängt. Im letzten Schritt wird die Instance dem neuen Cluster hinzugefügt.

Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb --region secondary-region-id \
  create-db-cluster \
    --db-cluster-identifier cluster-id \
    --global-cluster-identifier global-cluster-id \
    --engine-version version \
    --engine docdb

aws docdb --region secondary-region-id \
  create-db-instance \
    --db-cluster-identifier cluster-id \
    --global-cluster-identifier global-cluster-id \
    --engine-version version \
    --engine docdb
```

Für Windows:

```
aws docdb --region secondary-region-id ^
  create-db-cluster ^
    --db-cluster-identifier cluster-id ^
    --global-cluster-identifier global-cluster-id ^
    --engine-version version ^
    --engine docdb

aws docdb --region secondary-region-id ^
  create-db-instance ^
    --db-cluster-identifier cluster-id ^
    --global-cluster-identifier global-cluster-id ^
    --engine-version version ^
    --engine docdb
```

Verwenden eines Snapshots für Ihren globalen Amazon DocumentDB-Cluster

Sie können einen Snapshot eines Amazon DocumentDB-Clusters wiederherstellen, um ihn als Ausgangspunkt für Ihren globalen Cluster zu verwenden. Dazu müssen Sie den Snapshot wiederherstellen und einen neuen Cluster erstellen. Dies dient als primärer Cluster Ihres globalen

Clusters. Anschließend können Sie dem wiederhergestellten Cluster eine weitere Region hinzufügen und so in einen globalen Cluster umwandeln.

Verwalten eines globalen Amazon DocumentDB-Clusters

Sie führen die meisten Verwaltungsvorgänge auf den einzelnen Clustern durch, aus denen ein globaler Cluster besteht. Wenn Sie auf der Seite Cluster in der Konsole Gruppenbezogene Ressourcen auswählen, werden der primäre Cluster und die sekundären Cluster unter dem zugehörigen globalen Cluster gruppiert.

Auf der Registerkarte Konfiguration für einen globalen Cluster werden die angezeigt, AWS-Regionen in der die Cluster ausgeführt werden, die Version und die globale Cluster-ID.

Themen

Ändern eines globalen Amazon DocumentDB-Clusters

Auf der Seite Cluster in AWS Management Console der werden alle Ihre globalen Cluster aufgeführt, wobei der primäre Cluster und die sekundären Cluster für jeden Cluster angezeigt werden. Der globale Cluster hat seine eigenen Konfigurationseinstellungen. Insbesondere verfügt sie über Regionen, die ihren primären und sekundären Clustern zugeordnet sind.

Wenn Sie Änderungen am globalen Cluster vornehmen, haben Sie die Möglichkeit, Änderungen abzubrechen.

Wenn Sie Weiter wählen, bestätigen Sie die Änderungen.

Ändern von Parametern eines globalen Amazon DocumentDB-Clusters

Sie können die Cluster-Parametergruppen für jeden Cluster innerhalb des globalen Clusters unabhängig konfigurieren. Die meisten Parameter funktionieren genauso wie bei anderen Arten von Amazon DocumentDB-Clustern. Wir empfehlen, dass Sie die Einstellungen zwischen allen Clustern in einer globalen Datenbank konsistent halten. Dies hilft, unerwartete Verhaltensänderungen zu vermeiden, wenn Sie einen sekundären Cluster zum primären Cluster hochstufen.

Sie sollten z. B. die gleichen Einstellungen für Zeitzonen und Zeichensätze verwenden, um inkonsistentes Verhalten zu vermeiden, wenn ein anderer Cluster die Rolle des primären Clusters übernimmt.

Entfernen eines Clusters aus einem globalen Amazon DocumentDB-Cluster

Es gibt mehrere Situationen, in denen Sie Cluster möglicherweise aus Ihrem globalen Cluster entfernen möchten. Sie können beispielsweise einen Cluster aus einem globalen Cluster entfernen, wenn der primäre Cluster beeinträchtigt oder isoliert wird. Es wird dann zu einem eigenständigen bereitgestellten Cluster, der zum Erstellen eines neuen globalen Clusters verwendet werden könnte. Weitere Informationen finden Sie unter Manuelles Wiederherstellen eines globalen Clusters nach einem ungeplanten Ausfall.

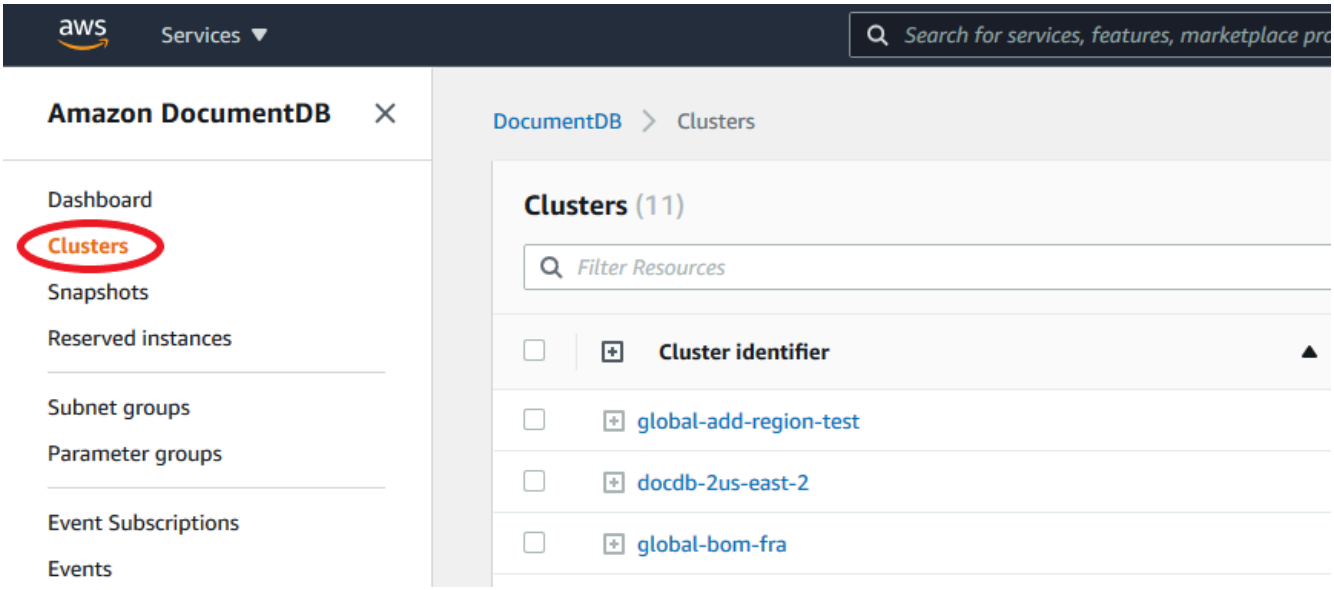
Möglicherweise möchten Sie auch Cluster entfernen, da Sie einen globalen Cluster löschen möchten, den Sie nicht mehr benötigen. Sie können den globalen Cluster erst löschen, nachdem Sie alle zugehörigen Cluster getrennt haben und den primären Cluster für den letzten Cluster verlassen haben. Weitere Informationen finden Sie unter Löschen eines globalen Amazon DocumentDB-Clusters.

> **Note**
>
> Wenn ein Cluster vom globalen Cluster getrennt wird, wird er nicht mehr mit dem primären Cluster synchronisiert. Es wird zu einem eigenständigen bereitgestellten Cluster mit vollständigen Lese-/Schreibfunktionen. Darüber hinaus ist es in der Amazon DocumentDB-Konsole nicht mehr sichtbar. Sie ist nur sichtbar, wenn Sie die Region in der Konsole auswählen, in der sich der Cluster befand.

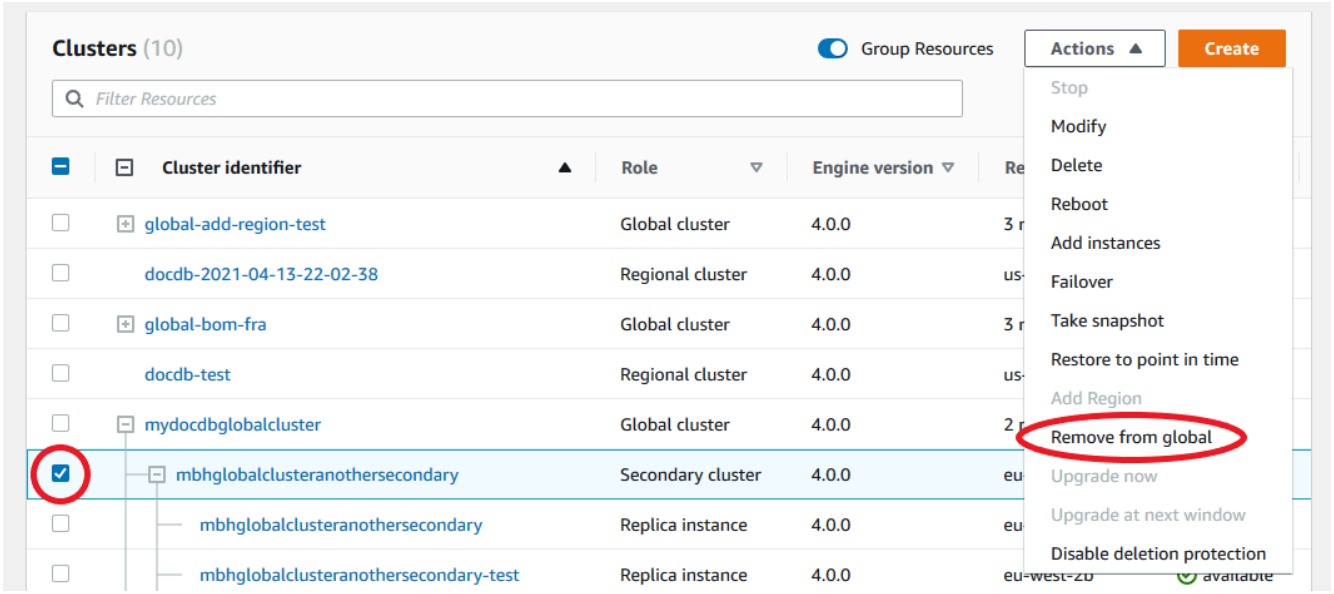
Sie können Cluster aus Ihrem globalen Cluster mithilfe der AWS Management Console, der AWS CLIoder der RDS-API entfernen.

Using the AWS Management Console

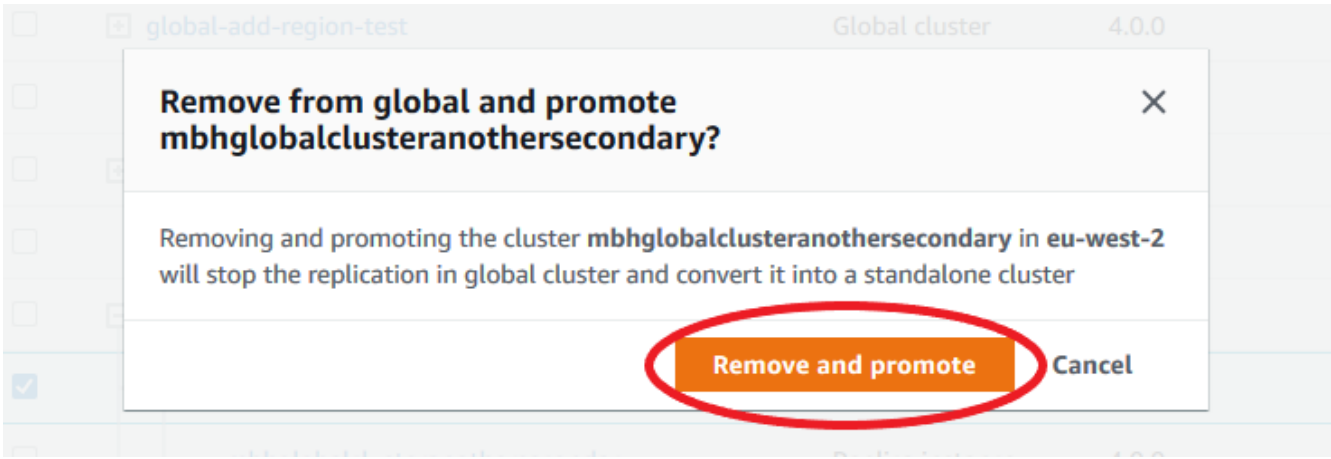
1. Melden Sie sich bei der an AWS Management Console und navigieren Sie zur Amazon DocumentDB-Konsole.
2. Wählen Sie auf der linken Seite der Navigation Cluster aus.



3. Erweitern Sie den globalen Cluster, sodass Sie alle sekundären Cluster sehen können. Wählen Sie die sekundären Cluster aus, die Sie entfernen möchten. Wählen Sie Aktionen und im Dropdown-Menü die Option Aus global entfernen aus.



4. Es wird eine Eingabeaufforderung angezeigt, in der Sie bestätigen müssen, dass Sie den sekundären Cluster vom globalen Cluster trennen möchten. Wählen Sie Entfernen und hochstufen, um den Cluster aus dem globalen Cluster zu entfernen.



Jetzt dient dieser Cluster nicht mehr als sekundärer Cluster und wird nicht mehr mit dem primären Cluster synchronisiert. Es handelt sich um einen eigenständigen Cluster mit voller Lese-/Schreibfähigkeit.

Nachdem Sie alle sekundären Cluster entfernt oder gelöscht haben, können Sie den primären Cluster auf die gleiche Weise entfernen. Sie können den primären Cluster erst vom globalen Cluster trennen oder entfernen, nachdem Sie alle sekundären Cluster entfernt haben. Der globale Cluster bleibt möglicherweise in der Liste Cluster mit null Regionen und AZs . Sie können löschen, wenn Sie diesen globalen Cluster nicht mehr verwenden möchten.

Using the AWS CLI

Um einen Cluster aus einem globalen Cluster zu entfernen, führen Sie den `remove-from-global-cluster` CLI-Befehl mit den folgenden Parametern aus:

- `--global-cluster-identifier` – Der Name (Bezeichner) Ihres globalen Clusters.
- `--db-cluster-identifier` – Der Name jedes Clusters, der aus dem globalen Cluster entfernt werden soll.

Die folgenden Beispiele entfernen zuerst einen sekundären Cluster und dann den primären Cluster aus einem globalen Cluster.

Für Linux, macOS oder Unix:

```
aws docdb --region secondary_region \
  remove-from-global-cluster \
    --db-cluster-identifier secondary_cluster_ARN \
```

```
    --global-cluster-identifier global_cluster_id

aws docdb --region primary_region \
  remove-from-global-cluster \
    --db-cluster-identifier primary_cluster_ARN \
    --global-cluster-identifier global_cluster_id
```

Wiederholen Sie den `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` Befehl für jede sekundäre Region in Ihrem globalen Cluster.

Für Windows:

```
aws docdb --region secondary_region ^
  remove-from-global-cluster ^
    --db-cluster-identifier secondary_cluster_ARN ^
    --global-cluster-identifier global_cluster_id

aws docdb --region primary_region ^
  remove-from-global-cluster ^
    --db-cluster-identifier primary_cluster_ARN ^
    --global-cluster-identifier global_cluster_id
```

Wiederholen Sie den `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` Befehl für jede sekundäre Region in Ihrem globalen Cluster.

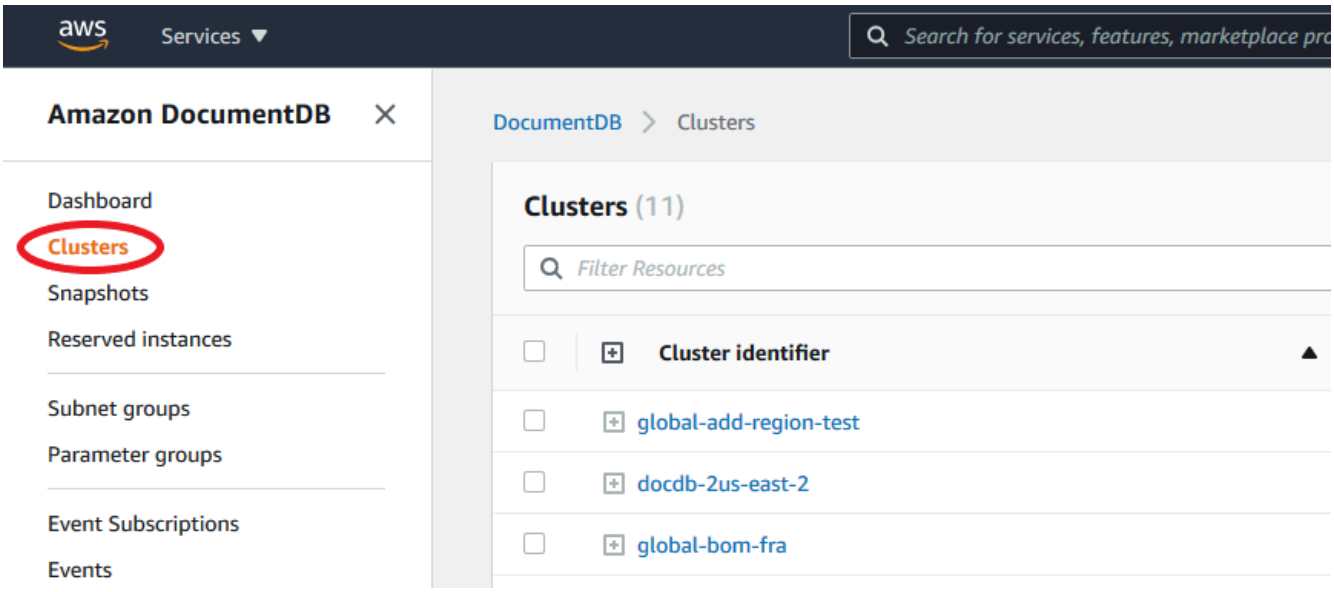
Löschen eines Clusters aus einem globalen Amazon DocumentDB-Cluster

Gehen Sie wie folgt vor, um einen globalen Cluster zu löschen:

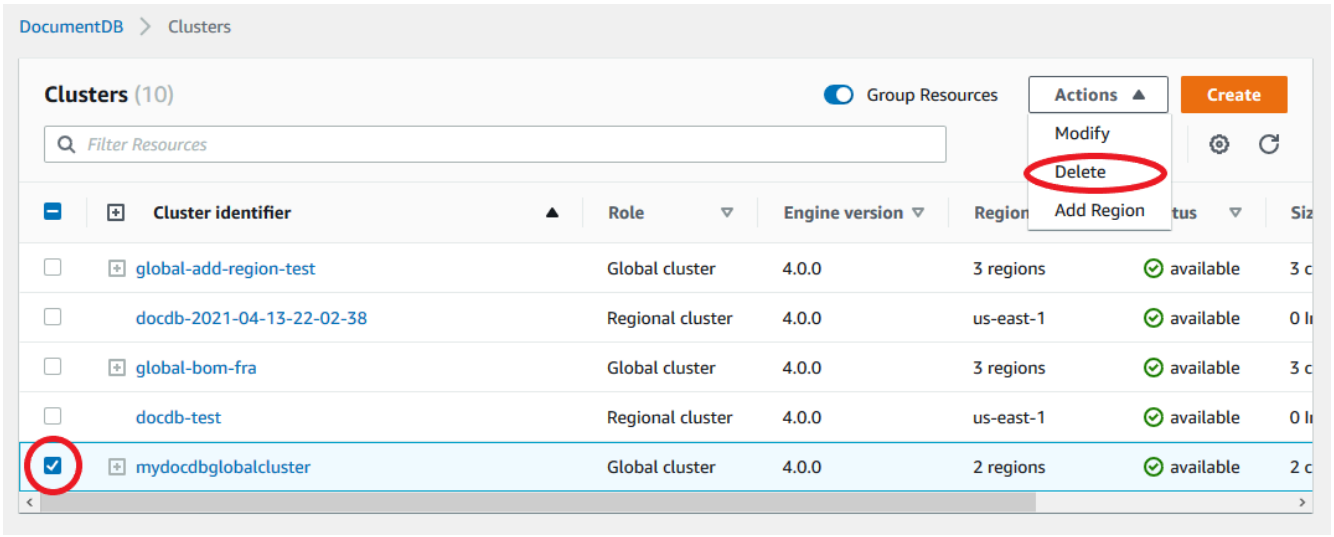
- Entfernen Sie alle sekundären Cluster aus dem globalen Cluster. Jeder Cluster wird zu einem eigenständigen Cluster. Weitere Informationen finden Sie im vorherigen Abschnitt zum Entfernen globaler Cluster.
- Löschen Sie aus jedem eigenständigen Cluster alle Replikate.
- Entfernen Sie den primären Cluster aus dem globalen Cluster. Dies wird zu einem eigenständigen Cluster.
- Löschen Sie zuerst alle Replikate aus dem primären Cluster und dann die primäre Instance. Durch das Löschen der primären Instance aus dem neu eigenständigen Cluster werden in der Regel sowohl der Cluster als auch der globale Cluster entfernt.

Using the AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und navigieren Sie zur Amazon DocumentDB-Konsole.
2. Wählen Sie Cluster und suchen Sie den globalen Cluster, den Sie löschen möchten.



3. Wählen Sie im Menü Aktionen die Option Löschen aus, wenn Ihr globaler Cluster ausgewählt ist.



Vergewissern Sie sich, dass alle Cluster aus dem globalen Cluster entfernt wurden. Der globale Cluster sollte null Regionen und AZs sowie eine Größe von null Clustern anzeigen. Wenn der globale Cluster Cluster Cluster enthält, können Sie ihn noch nicht löschen. Sie müssen zunächst die Anweisungen im vorherigen Schritt, Entfernen globaler Cluster, befolgen.

Using the AWS CLI

Um einen globalen Cluster zu löschen, führen Sie den `delete-global-cluster` CLI-Befehl mit dem Namen des AWS-Region und der globalen Cluster-ID aus, wie im folgenden Beispiel gezeigt.

Für Linux, macOS oder Unix:

```
aws docdb --region primary_region delete-global-cluster \
   --global-cluster-identifier global_cluster_id
```

Für Windows:

```
aws docdb --region primary_region delete-global-cluster ^
   --global-cluster-identifier global_cluster_id
```

Erstellen eines HeadlessAmazon DocumentDB-Clusters in einer sekundären Region

Obwohl ein globaler Amazon DocumentDB-Cluster mindestens einen sekundären Cluster in einem anderen AWS-Region als dem primären Cluster benötigt, können Sie eine Headless-Konfiguration für den sekundären Cluster verwenden. Ein sekundärer Amazon DocumentDB-Headless-Cluster ist ein Cluster ohne Instance. Diese Art der Konfiguration kann die Ausgaben für einen globalen Cluster senken. In einem Amazon DocumentDB-Cluster sind Rechenleistung und Speicher entkoppelt. Ohne die Instance wird Ihnen keine Rechenleistung in Rechnung gestellt, sondern nur der Speicher. Wenn es korrekt eingerichtet ist, wird das sekundäre Headless-Speichervolumen mit dem primären Cluster synchronisiert.

Sie fügen den sekundären Cluster wie gewohnt hinzu, wenn Sie einen globalen Amazon DocumentDB-Cluster erstellen. Nachdem der primäre Cluster jedoch mit der Replikation zum sekundären Cluster begonnen hat, löschen Sie die schreibgeschützte Instance aus dem sekundären Cluster. Dieser sekundäre Cluster gilt jetzt als „Headless“, da er keine Instance mehr hat. Das Speichervolumen wird jedoch mit dem primären Amazon DocumentDB-Cluster synchronisiert.

> **Important**
>
> Wir empfehlen nur Headless-Cluster für Kunden, die regionsweite Ausfälle für mehr als 15 Minuten tolerieren können. Dies liegt daran, dass der Benutzer nach einem regionsweiten Ausfall mit einem sekundären Headless-Cluster nach dem Failover eine neue Instance erstellen muss. Es kann ~10–15 Minuten dauern, bis eine neue Instance verfügbar ist.

So fügen Sie Ihrem globalen Cluster einen sekundären Headless-Cluster hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole .
2. Wählen Sie auf der linken Seite der Navigation Cluster aus.
3. Wählen Sie den globalen Cluster aus, der einen sekundären Cluster benötigt. Stellen Sie sicher, dass der primäre Cluster ist`Available`.
4. Wählen Sie unter Actions (Aktionen) die Option Add region (Region hinzufügen) aus.
5. Wählen Sie auf der Seite Region hinzufügen die sekundäre Region aus.

   > **Note**
   >
   > Sie können keine Region auswählen, die bereits über einen sekundären Cluster für denselben globalen Cluster verfügt. Außerdem kann es nicht dieselbe Region sein wie der primäre Cluster.

6. Füllen Sie die verbleibenden Felder für den sekundären Cluster in der neuen Region aus. Dies sind die gleichen Konfigurationsoptionen wie für jede Cluster-Instance.
7. Fügen Sie eine Region hinzu. Nachdem Sie die Region zu Ihrem globalen Cluster hinzugefügt haben, sehen Sie sie in der Liste von `Clusters` im AWS Management Console.
8. Überprüfen Sie den Status des sekundären Clusters und seiner Reader-Instance, bevor Sie fortfahren, indem Sie die AWS Management Console oder die verwendenAWS CLI. Hier ist ein Beispielbefehl, wenn Sie die verwendenAWS CLI:

   ```
   $ aws docdb describe-db-clusters --db-cluster-identifier secondary-cluster-id --
   query '*[].[Status]' --output text
   ```

   Es kann einige Minuten dauern, bis sich der Status eines neu hinzugefügten sekundären Clusters von „Erstellen“ zu „Verfügbar“ ändert. Wenn der Cluster verfügbar ist, können Sie die Reader-Instance löschen.
9. Wählen Sie die Reader-Instance im sekundären Cluster und dann Löschen aus.
10. Nach dem Löschen der Reader-Instance bleibt der sekundäre Cluster Teil des globalen Clusters. Es sollte keine Instance zugeordnet sein.

> **Note**
>
> Sie können diesen sekundären Amazon DocumentDB-Headless-Cluster verwenden, um Ihren globalen Amazon DocumentDB-Cluster nach einem ungeplanten Ausfall in der primären Region manuell wiederherzustellen, wenn ein solcher Ausfall auftritt.

Herstellen einer Verbindung mit einem Amazon DocumentDB Global Cluster

Wie Sie eine Verbindung zu einem globalen Cluster herstellen, hängt davon ab, ob Sie in den Cluster schreiben oder aus dem Cluster lesen müssen:

- Bei schreibgeschützten Anfragen oder Abfragen stellen Sie eine Verbindung zum Reader-Endpunkt für den Cluster in Ihrem herAWS-Region.
- Stellen Sie zum Ausführen von Data Manipulation Language (DML)- und Data Definition Language (DDL)-Anweisungen eine Verbindung zum Cluster-Endpunkt des primären Clusters her. Dieser Endpunkt befindet sich möglicherweise in einer anderen AWS-Region als Ihre Anwendung.

Wenn Sie einen globalen Cluster in der -Konsole anzeigen, können Sie alle Allzweck-Endpunkte sehen, die mit allen seinen Clustern verknüpft sind.

Wie Sie eine Verbindung zu einem globalen Cluster herstellen, hängt davon ab, ob Sie in die Datenbank schreiben oder aus der Datenbank lesen müssen. Für DDL-, DML- und Lesevorgänge, die Sie aus der primären Region bereitstellen möchten, sollten Sie eine Verbindung zu Ihrem primären Cluster herstellen. Wir empfehlen Ihnen, über den Cluster-Endpunkt im Replikatsatzmodus eine Verbindung zu Ihrem primären Cluster herzustellen, wobei die Lesepräferenz lautet`secondaryPreferred=true`. Dadurch wird Schreibdatenverkehr an die Writer-Instance Ihres primären Clusters und Lesedatenverkehr an die Replikat-Instance Ihres primären Clusters weitergeleitet.

Für regionsübergreifenden schreibgeschützten Datenverkehr sollten Sie eine Verbindung zu einem Ihrer sekundären Cluster herstellen. Wir empfehlen, dass Sie eine Verbindung zu Ihrem sekundären Cluster über den Cluster-Endpunkt im Replikat-Set-Modus herstellen. Da es sich bei allen Instances um schreibgeschützte Replikat-Instances handelt, müssen Sie keine Lesepräferenz angeben. Um die Latenz zu minimieren, wählen Sie den Reader-Endpunkt aus, der sich in Ihrer Region oder der Ihnen am nächsten gelegenen Region befindet.

Überwachen von globalen Amazon DocumentDB-Clustern

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist in integriert, CloudWatch sodass Sie Betriebsmetriken für Ihre Cluster sammeln und analysieren können. Sie können diese Metriken mithilfe der CloudWatch -Konsole, der Amazon DocumentDB-Konsole, der AWS Command Line Interface (AWS CLI) oder der CloudWatch -API überwachen.

Verwenden Sie die folgenden CloudWatch Metriken, um einen globalen Cluster zu überwachen.

Kennzahl	Beschreibung
`GlobalClusterReplicatedWriteIO`	Die durchschnittliche Anzahl von fakturierten Schreib-I/O-Operationen, die vom Cluster-V olume im primären AWS-Region zum Cluster-Volume in einem sekundären repliziert wurden AWS-Regionund in 5-Minuten-Intervallen gemeldet werden. Die Anzahl der `Replicate dWriteIOs` in jede sekundäre Region replizierten entspricht der Anzahl der In-Region en, die von der primären Region `VolumeWri teIOPs` ausgeführt werden.
`GlobalClusterDataTransferBytes`	Die Menge der Daten, die von der des primären Clusters AWS-Region in die eines sekundären Clusters übertragen werdenAWS-Region, wird in Byte gemessen.
`GlobalClusterReplicationLag`	Die Verzögerung in Millisekunden beim Replizieren von Änderungsereignissen vom primären Cluster AWS-Region zum sekundären Cluster AWS-Region

Weitere Informationen zum Anzeigen dieser Metriken finden Sie unter Anzeigen von CloudWatch Daten.

Notfallwiederherstellung und globale Amazon DocumentDB-Cluster

Durch die Verwendung eines globalen Clusters können Sie sich schnell nach Katastrophen wie Regionsausfällen wiederherstellen. Die Wiederherstellung nach einem Notfall wird in der Regel anhand von RTO- und RPO-Werten gemessen.

- Recovery Time Objective (RTO) – Die Zeit, die ein System benötigt, um nach einem Notfall in einen arbeitsfähigen Zustand zurückzukehren. Mit anderen Worten: RTO misst die Ausfallzeit. Für einen globalen Cluster kann RTO in der Reihenfolge von Minuten liegen.
- Recovery Point Objective (RPO) – Die Datenmenge, die verloren gehen kann (gemessen in Zeit). Für einen globalen Cluster wird RPO in der Regel in Sekunden gemessen.
- Um nach einem ungeplanten Ausfall eine Wiederherstellung durchzuführen, können Sie ein regionsübergreifendes Failover auf eines der sekundären Cluster in Ihrem globalen Cluster durchführen. Wenn Ihr globaler Cluster mehrere sekundäre Regionen hat, stellen Sie sicher, dass Sie alle sekundären Regionen trennen, wenn der primäre Cluster AWS-Region ausfällt. Anschließend stufen Sie eine dieser sekundären Regionen zum neuen primären hochAWS-Region. Schließlich erstellen Sie neue Cluster in jeder der anderen sekundären Regionen und fügen diese Cluster an Ihren globalen Cluster an.
- Wenn Sie einen sekundären Cluster zum primären Cluster hochstufen, müssen Sie auch die Endpunkte aktualisieren, die Ihre Anwendungen für die Verbindung mit dem globalen Cluster verwenden. Um einen neuen Schreiber-Endpunkt aus einem neu heraufgestuften Cluster zu erhalten, können Sie einen früheren Leser-Endpunkt konvertieren, indem Sie `-ro` aus der Endpunkt-Zeichenfolge entfernen. Wenn beispielsweise ein früherer Leser-Endpunkt `global-16rr-test-cluster-1.cluster-ro-12345678901.us-west-2.docdb.amazonaws.com` ist, dann ist der neue heraufgestufte Schreiber-Endpunkt `global-16rr-test-cluster-1.cluster-cps2igpwyrwa.us-west-2.rds.amazonaws.com`.

Failover für globale Amazon DocumentDB-Cluster

Wenn ein ganzer Cluster in einem nicht mehr verfügbar AWS-Region ist, können Sie einen anderen Cluster im globalen Cluster hochstufen, um Lese-/Schreibfähigkeit zu haben.

Sie können den Failover-Mechanismus manuell aktivieren, wenn ein Cluster in einer anderen die bessere Wahl AWS-Region ist, der primäre Cluster zu sein. Sie können beispielsweise die Kapazität eines sekundären Clusters erhöhen und diesen Cluster dann zum primären Cluster hochstufen. Oder das Aktivitätsgleichgewicht zwischen den AWS-Regionen kann sich ändern, sodass das Umschalten

des primären Clusters auf einen anderen zu einer geringeren Latenz für Schreibvorgänge führen AWS-Region kann.

Im folgenden Verfahren wird beschrieben, wie Sie einen der sekundären Cluster in einem globalen DocumentDB-Cluster heraufstufen.

So stufen Sie einen sekundären Cluster hoch:

1. Beenden Sie die Ausgabe von DML-Anweisungen und anderen Schreibvorgängen an den primären Cluster in der AWS-Region mit dem Ausfall.
2. Identifizieren Sie einen Cluster von einem sekundären ClusterAWS-Region, der als neuer primärer Cluster verwendet werden soll. Wenn Sie zwei (oder mehr) sekundäre AWS-Regionen in Ihrem globalen Cluster haben, wählen Sie den sekundären Cluster mit der geringsten Verzögerungszeit aus.
3. Trennen Sie den ausgewählten sekundären Cluster vom globalen Cluster.

   Das Entfernen eines sekundären Clusters aus einem globalen Cluster stoppt sofort die Replikation vom primären zu diesem sekundären Cluster und stuft ihn zu einem eigenständigen bereitgestellten Cluster mit vollständigen Lese-/Schreibfunktionen ein. Jeder andere sekundäre Cluster, der dem primären Cluster in der Region mit dem Ausfall zugeordnet ist, ist weiterhin verfügbar und kann Aufrufe von Ihrer Anwendung annehmen. Sie verbrauchen auch Ressourcen. Da Sie den globalen Cluster neu erstellen, entfernen Sie die anderen sekundären Cluster, bevor Sie den neuen globalen Cluster in den folgenden Schritten erstellen, um Split-Brain-Probleme und andere Probleme zu vermeiden.

   Ausführliche Schritte zum Trennen finden Sie unter Entfernen eines Clusters aus einem globalen Amazon DocumentDB-Cluster.
4. Konfigurieren Sie Ihre Anwendung neu, um alle Schreibvorgänge an diesen jetzt eigenständigen Cluster mit ihrem neuen Endpunkt zu senden. Wenn Sie die angegebenen Namen beim Erstellen des globalen Clusters akzeptiert haben, können Sie den Endpunkt ändern, indem Sie die -ro aus der Endpunktzeichenfolge des Clusters in Ihrer Anwendung entfernen.

   Beispielsweise `my-global.cluster-ro-aaaaaabbbbbb.us-west-1.docdb.amazonaws.com` wird der Endpunkt des sekundären Clusters zu , `my-global.cluster-aaaaaabbbbbb.us-west-1.docdb.amazonaws.com` wenn dieser Cluster vom globalen Cluster getrennt wird.

Dieser Cluster wird zum primären Cluster eines neuen globalen Clusters, wenn Sie im nächsten Schritt mit dem Hinzufügen von Regionen beginnen.

5. Fügen Sie AWS-Region dem Cluster ein hinzu. Wenn Sie dies tun, beginnt der Replikationsprozess vom primären zum sekundären Cluster.
6. Fügen Sie nach AWS-Regionen Bedarf weitere hinzu, um die Topologie neu zu erstellen, die zur Unterstützung Ihrer Anwendung erforderlich ist. Stellen Sie sicher, dass Anwendungsschreibvorgänge vor, während und nach Änderungen wie diesen an den richtigen Cluster gesendet werden, um Dateninkonsistenzen zwischen den Clustern im globalen Cluster zu vermeiden (Split-Brain-Probleme).
7. Wenn der Ausfall behoben ist und Sie bereit sind, Ihr Original erneut AWS-Region dem primären Cluster zuzuweisen, führen Sie dieselben Schritte in umgekehrter Reihenfolge aus.
8. Entfernen Sie einen der sekundären Cluster aus dem globalen Cluster. Auf diese Weise kann sie Lese-/Schreibdatenverkehr bereitstellen.
9. Leiten Sie den gesamten Schreibdatenverkehr an den primären Cluster im ursprünglichen umAWS-Region.
10. Fügen Sie ein hinzuAWS-Region, um einen oder mehrere sekundäre Cluster in derselben AWS-Region wie zuvor einzurichten.

Globale Amazon DocumentDB-Cluster können mit AWS-SDKs verwaltet werden, sodass Sie Lösungen zur Automatisierung des globalen Cluster-Failover-Prozesses für Anwendungsfälle zur Notfallwiederherstellung und Planung der Geschäftskontinuität erstellen können. SDKs Eine solche Lösung wird unseren Kunden im Rahmen der Apache-2.0-Lizenzierung zur Verfügung gestellt und kann über unser Tools-Repository hier aufgerufen werden. Diese Lösung nutzt Amazon Route53 für die Endpunktverwaltung und bietet AWS Lambda-Funktionen, die basierend auf geeigneten Ereignissen ausgelöst werden können.

Verwalten von Amazon DocumentDB-Clustern

Um einen Amazon DocumentDB-Cluster zu verwalten, benötigen Sie eine IAM-Richtlinie mit den entsprechenden Berechtigungen auf Steuerebene von Amazon DocumentDB. Diese Berechtigungen ermöglichen das Erstellen, Ändern und Löschen von Clustern und Instances. Die `AmazonDocDBFullAccess` Richtlinie stellt alle erforderlichen Berechtigungen für die Verwaltung eines Amazon DocumentDB-Clusters bereit.

Die folgenden Themen zeigen, wie Sie verschiedene Aufgaben ausführen, wenn Sie mit Amazon DocumentDB-Clustern arbeiten, einschließlich Erstellen, Löschen, Ändern, Verbinden mit und Anzeigen von Clustern.

Themen

Grundlegendes zu Clustern

Amazon DocumentDB trennt Datenverarbeitung und -speicher und entlädt die Datenreplikation und -sicherung auf das Cluster-Volume. Ein Cluster-Volume bietet eine dauerhafte, zuverlässige und hochverfügbare Speicherschicht, die Daten auf sechs Arten über drei Availability Zones repliziert. Replikate ermöglichen eine höhere Datenverfügbarkeit und Leseskalierung. Jeder Cluster kann bis zu 15 Replikate skalieren.

Substantiv	Beschreibung	API-Operationen (Verben)

Cluster	Besteht aus einer oder mehreren Instances und einem Cluster-Storage-Volume, das die Daten für diese Instances verwaltet.	`create-db-cluster`
		`delete-db-cluster`
		`describe-db-clusters`
		`modify-db-cluster`

Instance	Das Lesen und Schreiben von Daten auf das Cluster-Speichervolumen erfolgt über Instances. In einem	`create-db-instance` `delete-db-instance`

Substantiv	Beschreibung	API-Operationen (Verben)

	gegebenen Cluster gibt es zwei Arten von Instances : Primär und Replikat. Ein Cluster hat immer eine primäre Instance und kann 0–15 Replikate haben.	`describe-db-instances`
		`modify-db-instance`
		`describe-orderable-db-instance-options`
		`reboot-db-instance`

Cluster-Volumen	Ein virtuelles Datenbank speichervolumen, das sich über drei Availability Zones erstreckt, wobei jede Availability Zone zwei Kopien der Cluster-Daten aufweist.	N/A
Primäre Instance	Unterstützt Lese- und Schreiboperationen und führt alle Datenänderungen im Cluster-Volume durch. Jeder Cluster verfügt über eine primäre Instance.	N/A
Replikat-Instance	Unterstützt nur Lesevorgänge. Jeder Amazon DocumentDB-Cluster kann zusätzlich zur primären Instance bis zu 15 Replikat-Instances haben. Mehrere Replikate verteilen den Lese-Workload. Durch die Platzierung von Replikaten in separaten Availability Zones können Sie außerdem die Verfügbarkeit der Datenbank erhöhen.	N/A

Substantiv	Beschreibung	API-Operationen (Verben)
Cluster-Endpunkt	Ein Endpunkt für einen Amazon DocumentDB-Cluster , der eine Verbindung zur aktuellen primären Instance für den Cluster herstellt. Jeder Amazon DocumentDB-Cluster verfügt über einen Cluster-Endpunkt und eine primäre Instance.	N/A
Leser-Endpunkt	Ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zu einem der verfügbaren Replikate für diesen Cluster herstellt. Jeder Amazon DocumentDB-Cluster verfügt über einen Reader-Endpunkt. Wenn es mehr als ein Replikat gibt, leitet der Reader-Endpunkt jede Verbindungsanforderung an eines der Amazon DocumentDB-Replikate weiter.	N/A
Instance-Endpunkt	Ein Endpunkt für eine Instance in einem Amazon DocumentDB-Cluster, der eine Verbindung zu einer bestimmten Instance herstellt. Jede Instance in einem Cluster hat unabhängig vom Instance-Typ einen eigenen eindeutigen Instance-Endpunkt.	N/A

Amazon DocumentDB-Cluster-Einstellungen

Beim Erstellen oder Ändern eines Clusters müssen Sie wissen, welche Parameter unveränderlich und welche nach dem Anlegen des Clusters veränderbar sind. Die folgende Tabelle listet alle Einstellungen bzw. Parameter auf, die für einen Cluster spezifisch sind. Wie in der Tabelle angegeben, sind nur einige Parameter änderbar.

> **Note**
>
> Diese Einstellungen sollten nicht mit Amazon DocumentDB-Cluster-Parametergruppen und ihren Parametern verwechselt werden. Weitere Informationen zu Cluster-Parametergruppen finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

Parameter	Anpassbar	Hinweise
`DBClusterIdentifier`	Ja	Benennungseinschränkungen: • Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten. • Muss für alle Cluster in Amazon RDS, Amazon Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
`Engine`	Nein	Der Wert muss `docdb` sein.
`BackupRetentionPeriod`	Ja	Muss zwischen 1 und 35 Tagen liegen.
`DBClusterParameterGroupName`	Ja	Benennungseinschränkungen: • Die Länge beträgt [1–255] alphanumerische Zeichen. • Muss mit einem Buchstaben beginnen.

Parameter	Anpassbar	Hinweise
		• Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
DBSubnetGroupName	Nein	Sie können das Subnetz des Clusters nach dem Anlegen eines Clusters nicht mehr ändern.
EngineVersion	Nein	Der Wert kann `5.0.0` (Standard), `4.0.0`oder sein`3.6.0`.
KmsKeyId	Nein	Wenn Sie Ihren Cluster verschlüsseln möchten, können Sie den AWS KMS Schlüssel, den Sie zur Verschlüsselung Ihres Clusters verwendet haben, nicht ändern.
MasterUsername	Nein	Sie können den `MasterUsername` nach dem Anlegen eines Clusters nicht mehr ändern. Benennungseinschränkungen: • Die Länge beträgt [1–63] alphanumerische Zeichen. • Muss mit einem Buchstaben beginnen. • Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

MasterUserPassword	Ja	Einschränkungen:
		
		• Die Länge beträgt [8–100] druckbare ASCII-Zeichen.
		• Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
		  • **/** (Schrägstrich)
		  • **"** (doppeltes Anführungszeichen)
		  • **@** ('At'-Symbol)

Parameter	Anpassbar	Hinweise
`Port`	Ja	Die Portnummer gilt für alle Instances im Cluster.
`PreferredBackupWindow`	Ja	
`PreferredMaintenanceWindow`	Ja	
`StorageEncrypted`	Nein	Wenn Sie Ihren Cluster verschlüsseln, kann er nicht unverschlüsselt sein.
`StorageType`	Ja	Der Speichertyp für den DB-Cluster: Standard (`standard`) oder E/A-optimiert (`iopt1`). Standard: `standard` Dieser Parameter kann mit `CreateDBCluster` und konfiguriert werden`ModifyDBCluster`. Weitere Informationen finden Sie unter Speicherkonfigurationen für Amazon DocumentDB-Cluster.
`Tags`	Ja	
`VpcSecurityGroupIds`	Nein	Nachdem ein Cluster erstellt wurde, können Sie die VPC nicht ändern, in der sich der Cluster befindet.

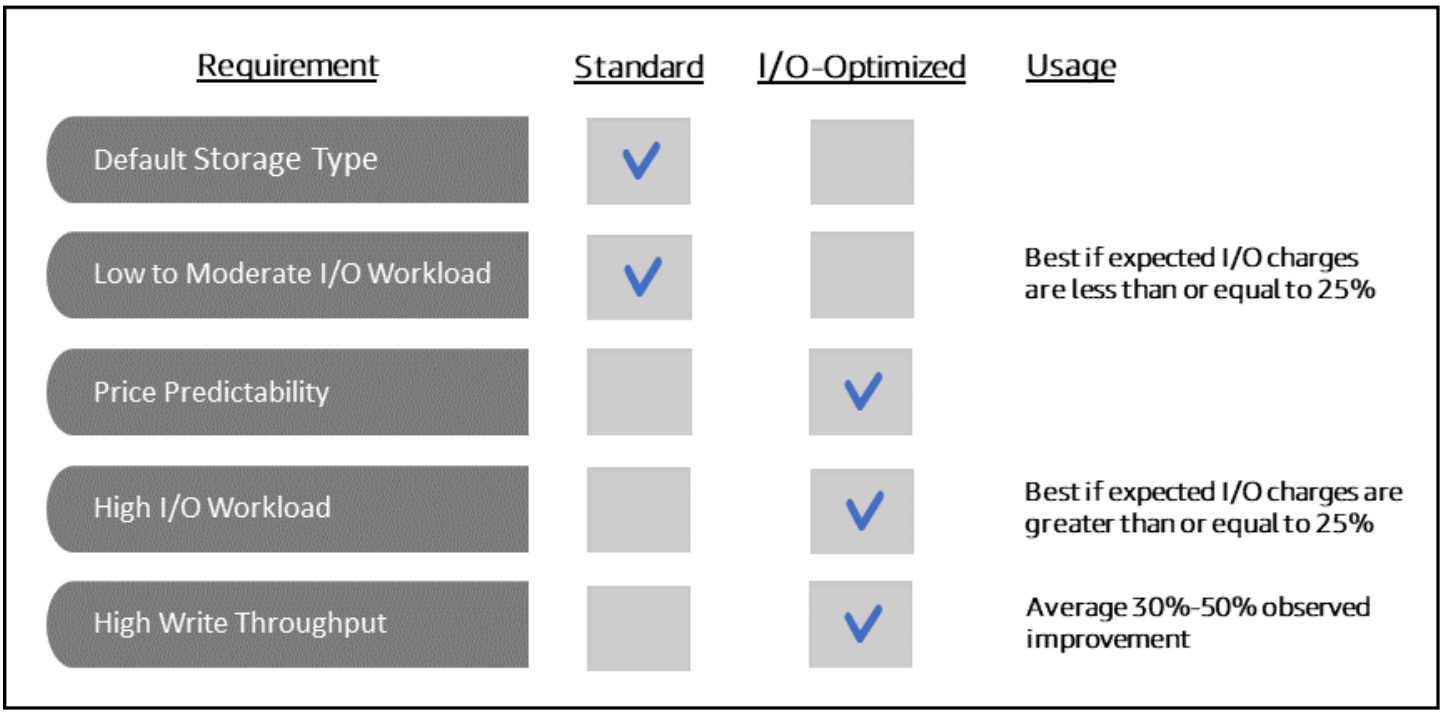
Speicherkonfigurationen für Amazon DocumentDB-Cluster

Ab Amazon DocumentDB 5.0 unterstützen Instance-basierte Cluster zwei Speicherkonfigurationstypen:

- Amazon DocumentDB-Standardspeicher: Konzipiert für Kunden mit geringem bis mittlerem E/A-Verbrauch. Wenn Sie davon ausgehen, dass Ihre E/A-Kosten weniger als 25 % Ihres gesamten Amazon DocumentDB-Clusters betragen, ist diese Auswahl möglicherweise ideal für Sie. Mit der Amazon DocumentDB-Standardspeicherkonfiguration werden Ihnen zusätzlich zu den Instance pay-per-request - und Speichergebühren E/A-Gebühren in Rechnung gestellt. Das bedeutet, dass Ihre Abrechnung je nach Nutzung von einem Zyklus zum anderen variieren kann. Die Konfiguration ist auf schwankende E/A-Anforderungen Ihrer Anwendung zugeschnitten.
- Amazon DocumentDB-E/A-optimierter Speicher: Konzipiert für Kunden, die Preisvorhersehbarkeit priorisieren oder über E/A-intensive Anwendungen verfügen. Die E/A-optimierte Konfiguration bietet eine verbesserte Leistung, einen höheren Durchsatz und eine geringere Latenz für Kunden mit E/A-intensiven Workloads. Wenn Sie davon ausgehen, dass Ihre I/O-Kosten 25 % Ihrer gesamten Amazon DocumentDB-Clusterkosten überschreiten, bietet diese Option eine verbesserte Preisleistung. Mit der Amazon DocumentDB-E/A-optimierten Speicherkonfiguration werden Ihnen keine Gebühren auf der Grundlage von E/A-Operationen berechnet, wodurch vorhersehbare Kosten in jedem Abrechnungszeitraum sichergestellt werden. Die Konfiguration stabilisiert die Kosten und verbessert gleichzeitig die Leistung.

Sie können Ihre vorhandenen Datenbank-Cluster einmal alle 30 Tage auf Amazon DocumentDB-E/A-optimierten Speicher umstellen. Sie können jederzeit wieder zum Amazon DocumentDB-Standardspeicher wechseln. Das nächste Datum, an dem die Speicherkonfiguration in E/A-optimiert geändert wird, kann mit dem `describe-db-clusters` Befehl über die AWS CLI oder über die AWS Management Console auf der Konfigurationsseite des Clusters verfolgt werden.

Sie können einen neuen Datenbank-Cluster erstellen, einschließlich der Amazon DocumentDB-E/A-optimierten Konfiguration, oder Ihre vorhandenen Datenbank-Cluster mit wenigen Klicks in der AWS Management Console, einer einzigen Parameteränderung in der AWS Command Line Interface (AWS CLI) oder über AWS SDKs konvertieren. Es gibt keine Ausfallzeiten oder Neustarts von Instances, die während oder nach der Änderung der Speicherkonfiguration erforderlich sind.

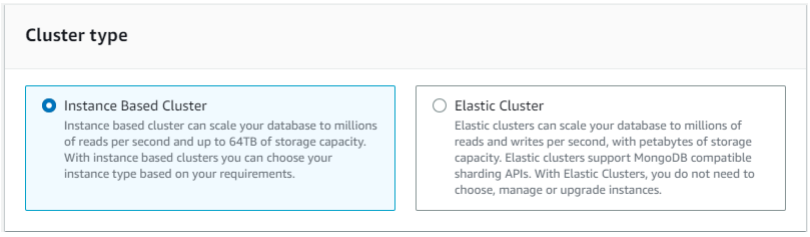


Erstellen eines E/A-optimierten Clusters

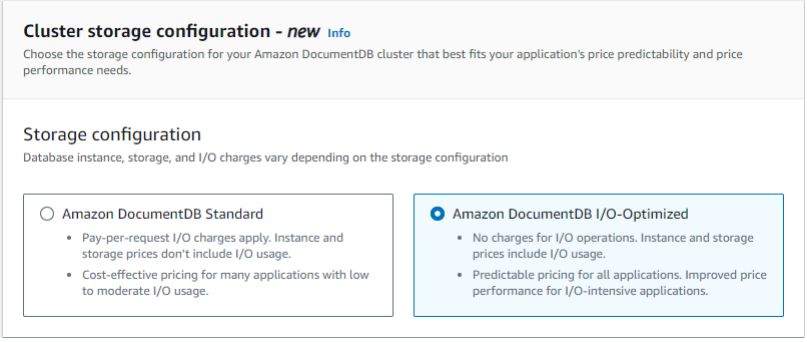
Using the AWS Management Console

So erstellen oder ändern Sie einen E/A-optimierten Cluster mithilfe der AWS Management Console:

1. Wählen Sie in der Amazon DocumentDB-Managementkonsole unter Cluster entweder Cluster erstellen oder den Cluster und dann Aktionen und dann Ändern aus.
2. Wenn Sie einen neuen Cluster erstellen, stellen Sie sicher, dass Sie Instance-basierte Cluster im Abschnitt Cluster-Typ auswählen (dies ist die Standardoption).



3. Wählen Sie im Abschnitt Konfiguration unter Cluster-Speicherkonfiguration die Option Amazon DocumentDB-E/A-optimiert aus.



4. Schließen Sie Ihre Cluster-Erstellung oder -Änderung ab und wählen Sie Cluster erstellen oder Cluster ändern aus.

   Den vollständigen Prozess zum Erstellen eines Clusters finden Sie unter Erstellen eines Clusters und einer primären Instance mithilfe der AWS Management Console.

   Den vollständigen Vorgang zum Ändern des Clusters finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Using the AWS CLI

So erstellen Sie einen E/A-optimierten Cluster mit der AWS CLI:

Ersetzen Sie in den folgenden Beispielen alle *`Platzhalter für Benutzereingabe`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 5.0.0 \
      --storage-type iopt1 \
      --deletion-protection \
      --master-username username \
      --master-user-password password
```

Für Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
```

```
    --engine-version 5.0.0 ^
    --storage-type iopt1 ^
    --deletion-protection ^
    --master-username username ^
    --master-user-password password
```

Kostenanalyse zur Bestimmung der Speicherkonfiguration

Mit Amazon DocumentDB haben Sie die Flexibilität, Ihre Speicherkonfiguration für jeden Datenbank-Cluster auszuwählen, den Sie haben. Um Ihre Cluster ordnungsgemäß zwischen Standard- und E/A-optimierten zuzuweisen, können Sie Ihre Amazon DocumentDB-Kosten clusterweise verfolgen. Dazu können Sie Tags zu vorhandenen Clustern hinzufügen, die Kostenzuordnungsmarkierung in Ihrem AWS Billing and Cost Management Dashboard aktivieren und Ihre Kosten für einen bestimmten Cluster in der analysierenAWS Cost Explorer Service. Weitere Informationen zur Kostenanalyse finden Sie in unserem Blog Verwenden von Kostenzuordnungs-Tags.

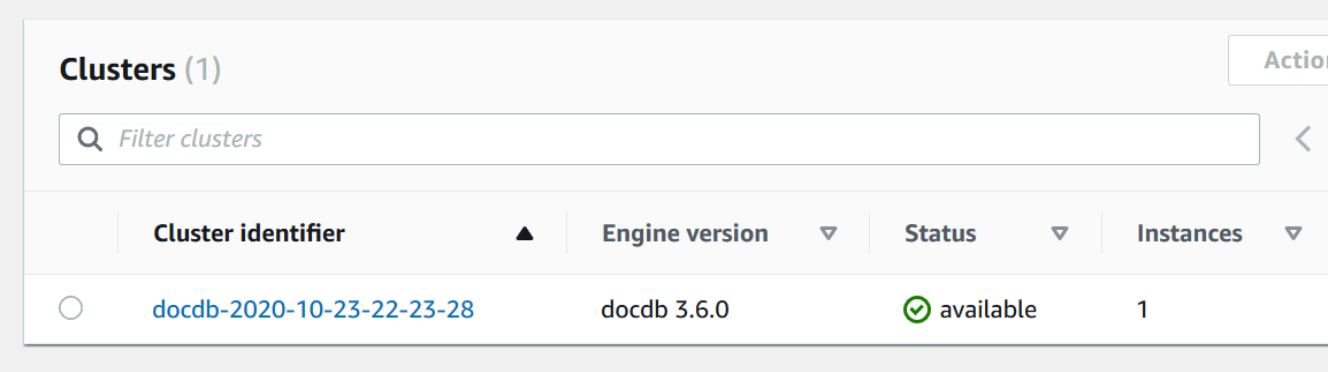
Ermitteln des Cluster-Status

Sie können den Status eines Clusters mithilfe der AWS Management Console oder der bestimmen AWS CLI.

Using the AWS Management Console

Gehen Sie wie folgt vor, um den Status Ihres Amazon DocumentDB-Clusters mithilfe der anzuzeigen. AWS Management Console

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Suchen Sie in der Spalte Cluster identifier (Cluster-ID) den Namen des Clusters, der Sie interessiert. Um dann den Status des Clusters zu ermitteln, lesen Sie diese Zeile in die Spalte Status, wie unten gezeigt.



Using the AWS CLI

Verwenden Sie die -`describe-db-clusters`Operation, um den Status Ihres Amazon DocumentDB-Clusters mithilfe der anzuzeigen AWS CLI.

Der folgende Code ermittelt den Status des Clusters `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Amazon DocumentDB-Cluster-Lebenszyklus

Der Lebenszyklus eines Amazon DocumentDB-Clusters umfasst das Erstellen, Beschreiben, Ändern und Löschen des Clusters. Dieser Abschnitt enthält Informationen darüber, wie Sie diese Prozesse abschließen können.

Themen

Erstellen eines Amazon DocumentDB-Clusters

Ein Amazon DocumentDB-Cluster besteht aus Instances und einem Cluster-Volume, das die Daten für den Cluster darstellt. Das Cluster-Volumen wird auf sechs Arten über drei Availability Zones als ein einziges, virtuelles Volume repliziert. Der Cluster enthält eine primäre Instance und optional bis zu 15 Replikat-Instances.

In den folgenden Abschnitten wird gezeigt, wie Sie einen Amazon DocumentDB-Cluster mithilfe der AWS Management Console oder der erstellen AWS CLI. Sie können dann weitere Replikat-Instances für diesen Cluster hinzufügen. Wenn Sie die Konsole verwenden, um Ihren Amazon DocumentDB-Cluster zu erstellen, wird automatisch gleichzeitig eine primäre Instance für Sie erstellt. Wenn Sie die verwenden, AWS CLI um Ihren Amazon DocumentDB-Cluster zu erstellen, müssen Sie, nachdem der Status des Clusters verfügbar ist, die primäre Instance für diesen Cluster erstellen.

Voraussetzungen

Im Folgenden sind die Voraussetzungen für die Erstellung eines Amazon DocumentDB-Clusters aufgeführt.

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um eines zu erstellen.

So registrieren Sie sich für ein AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für ein registrieren AWS-Konto, Root-Benutzer des AWS-Kontos wird ein erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

VPC-Voraussetzungen

Sie können einen Amazon DocumentDB-Cluster nur in einer Amazon Virtual Private Cloud (Amazon VPC) erstellen. Ihre Amazon VPC muss mindestens ein Subnetz in jeder von mindestens zwei Availability Zones haben, damit Sie sie mit einem Amazon DocumentDB-Cluster verwenden können. Durch die Verteilung Ihrer Cluster-Instances auf Availability Zones stellen Sie sicher, dass Instances in Ihrem Cluster verfügbar sind, falls es sich nicht um einen Ausfall der Availability Zone handelt.

Voraussetzungen für Subnetze

Beim Erstellen eines Amazon DocumentDB-Clusters müssen Sie eine VPC und eine entsprechende Subnetzgruppe innerhalb dieser VPC auswählen, um Ihren Cluster zu starten. Subnetze bestimmen die Availability Zone und den IP-Bereich innerhalb dieser Availability Zone, die Sie zum Starten einer Instance verwenden möchten. Im Rahmen dieser Ausführungen werden die Begriffe Subnetz und Availability Zone synonym verwendet. Eine Subnetzgruppe ist ein benannter Satz von Subnetzen (oder Availability Zones). Sie können mit einer Subnetzgruppe die Availability Zones angeben, die Sie zum Starten von Amazon DocumentDB-Instances verwenden möchten. So wird beispielsweise in einem Cluster mit drei Instances für eine hohe Verfügbarkeit empfohlen, dass jede dieser Instances in separaten Availability Zones bereitgestellt wird. Wenn eine einzelne Availability Zones ausfällt, wirkt sich dies nur auf eine einzelne Instance aus.

Amazon DocumentDB-Instances können derzeit in bis zu drei Availability Zones bereitgestellt werden. Selbst wenn eine Subnetzgruppe über mehr als drei Subnetze verfügt, können Sie nur drei dieser Subnetze verwenden, um einen Amazon DocumentDB-Cluster zu erstellen. Daher wird empfohlen, dass Sie beim Anlegen einer Subnetzgruppe nur die drei Subnetze auswählen, in denen

Sie Ihre Instances bereitstellen möchten. In USA Ost (Nord-Virginia) kann Ihre Subnetzgruppe sechs Subnetze (oder Availability Zones) haben. Wenn jedoch ein Amazon DocumentDB-Cluster bereitgestellt wird, wählt Amazon DocumentDB drei dieser Availability Zones aus, die für die Bereitstellung von Instances verwendet werden.

Angenommen, Amazon DocumentDB wählt beim Erstellen eines Clusters die Availability Zones {1A, 1B und 1C } aus. Wenn Sie nun versuchen, eine Instance in Availability Zone {1D} zu erstellen, schlägt der API-Aufruf fehl. Wenn Sie jedoch eine Instance erstellen möchten, ohne eine bestimmte Availability Zone anzugeben, wählt Amazon DocumentDB in Ihrem Namen eine Availability Zone aus. Amazon DocumentDB verwendet einen Algorithmus, um die Instances über Availability Zones hinweg auszugleichen, um eine hohe Verfügbarkeit zu erreichen. Wenn beispielsweise drei Instances bereitgestellt werden, werden diese standardmäßig über drei Availability Zones verteilt und nicht alle in einer einzigen Availability Zone bereitgestellt.

Empfehlungen:

- Wenn Sie keinen speziellen Grund haben, legen Sie immer eine Subnetzgruppe mit drei Subnetzen an. Auf diese Weise wird sichergestellt, dass Cluster mit drei oder mehr Instances eine höhere Verfügbarkeit erreichen können, da Instances über drei Availability Zones bereitgestellt werden.
- Verteilen Sie Instances immer über mehrere Availability Zones, um eine hohe Verfügbarkeit zu erreichen. Platzieren Sie niemals alle Instances für einen Cluster in einer einzigen Availability Zone.
- Da Failover-Ereignisse jederzeit auftreten können, sollten Sie nicht davon ausgehen, dass sich eine primäre Instance oder Replikat-Instance immer in einer bestimmten Availability Zone befinden.

Zusätzliche Voraussetzungen

Im Folgenden finden Sie einige zusätzliche Voraussetzungen für die Erstellung eines Amazon DocumentDB-Clusters:

- Wenn Sie eine Verbindung zu AWS mithilfe von AWS Identity and Access Management (IAM)-Anmeldeinformationen herstellen, muss Ihr IAM-Konto über die IAM-Richtlinien verfügen, die die erforderlichen Berechtigungen zum Ausführen von Amazon DocumentDB-Operationen gewähren.

  Wenn Sie ein IAM-Konto verwenden, um auf die Amazon DocumentDB-Konsole zuzugreifen, müssen Sie sich zuerst bei der AWS Management Console mit Ihrem IAM-Konto anmelden. Rufen Sie dann die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb auf.
- Wenn Sie die Konfigurationsparameter für Ihren Cluster anpassen möchten, müssen Sie eine Cluster-Parametergruppe und eine Parametergruppe mit den erforderlichen

Parametereinstellungen festlegen. Informationen zum Erstellen oder Ändern einer Cluster-Parametergruppe oder Parametergruppe finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

- Sie müssen die TCP/IP-Portnummer festlegen, die Sie für Ihren Cluster angeben möchten. Die Firewalls einiger Unternehmen blockieren Verbindungen zu den Standardports für Amazon DocumentDB . Wenn die Firewall Ihres Unternehmens den Standard-Port blockiert, wählen Sie einen anderen Port für Ihr Cluster aus. Alle Instances in einem Cluster verwenden denselben Port.

Erstellen eines Clusters und einer primären Instance mithilfe der AWS Management Console

In den folgenden Verfahren wird beschrieben, wie Sie die Konsole verwenden, um einen Amazon DocumentDB-Cluster mit einer oder mehreren Instances zu starten.

Erstellen eines Clusters: unter Verwendung von Standardeinstellungen

So erstellen Sie einen Cluster mit Instances unter Verwendung der Standardeinstellungen mithilfe der AWS Management Console

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wenn Sie Ihren Cluster in einer AWS-Region anderen als der Region USA Ost (Nord-Virginia) erstellen möchten, wählen Sie die Region aus der Liste im oberen rechten Bereich der Konsole aus.
3. Wählen Sie im Navigationsbereich Clusters (Cluster) und dann Create (Erstellen).

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

4. Füllen Sie auf der Seite Amazon DocumentDB-Cluster erstellen den Bereich Konfiguration aus.

   a. Cluster-ID – Akzeptieren Sie den von Amazon DocumentDB bereitgestellten Namen oder geben Sie einen Namen für Ihren Cluster ein, z. B. **`sample-cluster`**.

      Einschränkungen bei der Benennung von Clustern:

- Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
- Muss mit einem Buchstaben beginnen.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.

b. Engine-Version – Akzeptieren Sie die Standard-Engine-Version von 4.0.0 oder wählen Sie optional 3.6.0.

c. Instance-Klasse – Akzeptieren Sie die Standard- `db.r5.large`oder wählen Sie die gewünschte Instance-Klasse aus der Liste aus.

d. Anzahl der Instances – Wählen Sie in der Liste die Anzahl der Instances aus, die mit diesem Cluster erstellt werden sollen. Die erste Instance ist die primäre Instance. Alle anderen Instances sind schreibgeschützte Replikat-Instances. Sie können später Instances hinzufügen und löschen. Standardmäßig wird ein Amazon DocumentDB-Cluster mit drei Instances (ein Primär- und zwei Replikaten) gestartet.

5. Füllen Sie den Abschnitt Cluster-Speicherkonfiguration aus.

   Wählen Sie entweder Amazon DocumentDB Standard (Standard) oder Amazon DocumentDB I/O-optimized aus. Weitere Informationen finden Sie unter Speicherkonfigurationen für Amazon DocumentDB-Cluster.

6. Vervollständigen Sie den Bereich Authentication (Authentifizierung).

   a. Hauptbenutzername – Geben Sie einen Namen für den Hauptbenutzer ein. Um sich bei Ihrem Cluster anzumelden, müssen Sie den Hauptbenutzernamen verwenden.

      Benennungseinschränkungen für Master-Benutzer:

      - Die Länge beträgt [1–63] alphanumerische Zeichen.
      - Muss mit einem Buchstaben beginnen.
      - Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

   b. Hauptpasswort – Geben Sie ein Passwort für den Hauptbenutzer ein und bestätigen Sie es dann. Um sich bei Ihrem Cluster anzumelden, müssen Sie das Hauptpasswort verwenden.

      Einschränkungen für Master-Passwort:

      - Länge beträgt [8–100] druckbare ASCII-Zeichen.

- Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
  - **/** (Schrägstrich)
  - **"** (doppeltes Anführungszeichen)
  - **@** ('At'-Symbol)

7. Wählen Sie am unteren Rand der Seite eine der folgenden Optionen:
   - Um den Cluster jetzt zu erstellen, wählen Sie Create Cluster (Cluster erstellen) aus.
   - Wenn der Cluster nicht neu erstellt werden soll, wählen Sie Cancel (Abbrechen) aus.
   - Um den Cluster vor der Erstellung weiter zu konfigurieren, wählen Sie Show additional configurations (Weitere Konfigurationen anzeigen) aus und fahren dann mit Erstellen eines Clusters: zusätzliche Konfigurationen fort.

     Im Bereich Additional Configurations (Zusätzliche Konfigurationen) sind die folgenden Konfigurationen zu finden:

     - Netzwerkeinstellungen – Standardmäßig wird die `default` VPC-Sicherheitsgruppe verwendet.
     - Cluster-Optionen – Die Standardeinstellung ist die Verwendung von Port 27017 und der Standardparametergruppe.
     - Verschlüsselung – Standardmäßig wird die Verschlüsselung mit dem `-(default) aws/rds`Schlüssel aktiviert.

       > **Important**
       >
       > Die Verschlüsselung eines einmal verschlüsselten Clusters kann nicht wieder aufgehoben werden.

     - Backup – Standardmäßig werden Backups 1 Tag lang aufbewahrt und Amazon DocumentDB das Backup-Fenster auswählen lassen.
     - Protokollexporte – Standardmäßig werden keine Audit-Protokolle nach - CloudWatch Protokolle exportiert.
     - Wartung – Standardmäßig kann Amazon DocumentDB das Wartungsfenster auswählen.
     - Löschschutz – Schützen Sie Ihren Cluster vor versehentlichem Löschen. Cluster, die mit der Konsole erstellt wurden, sind standardmäßig aktiviert.

Wenn Sie die Standardeinstellungen jetzt übernehmen, können Sie die meisten davon später durch Bearbeitung des Clusters wieder ändern.

8. Aktivieren Sie eingehende Verbindungen für die Sicherheitsgruppe Ihres Clusters.

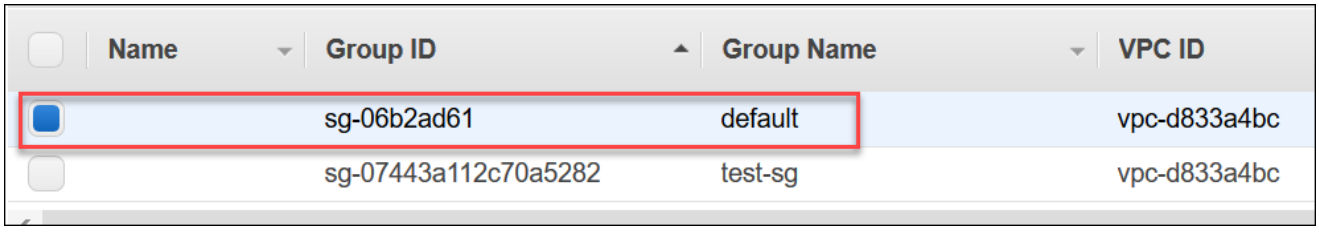
   Wenn Sie die Standardeinstellungen für Ihr Cluster nicht geändert haben, wurde mithilfe der Standard-Sicherheitsgruppe ein Cluster für die Standard-VPC in der angegebenen Region erstellt. Um eine Verbindung zu Amazon DocumentDB herzustellen, müssen Sie eingehende Verbindungen auf Port 27017 (oder dem Port Ihrer Wahl) für die Sicherheitsgruppe Ihres Clusters aktivieren.

   So fügen Sie eine eingehende Verbindung zur SicherheitsgruppeIhres Clusters hinzu:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Wählen Sie im Abschnitt Resources (Ressourcen) des Hauptfensters die Option Security groups (Sicherheitsgruppen).

   c. Wählen Sie in der Liste der Sicherheitsgruppen die Sicherheitsgruppe aus, die Sie bei der Erstellung Ihres Cluster verwendet haben (höchstwahrscheinlich handelt es sich dabei um die Standard-Sicherheitsgruppe), und wählen Sie das Feld links neben dem Name der Sicherheitsgruppe aus.



d. Wählen Sie im Menü Aktionen die Option Eingangsregeln bearbeiten aus. Wählen Sie dann die Regeleinschränkungen aus oder geben Sie diese ein.

   i. Typ – Wählen Sie aus der Liste das Protokoll aus, das für den Netzwerkverkehr geöffnet werden soll.

   ii. Protokoll – Wählen Sie aus der Liste den Protokolltyp aus.

   iii. Portbereich – Geben Sie für eine benutzerdefinierte Regel eine Portnummer oder einen Portbereich ein. Stellen Sie sicher, dass die Portnummer bzw. der Portbereich den Port enthält, den Sie beim Erstellen des Clusters angegeben haben (Standard: 27017).

   iv. Quelle – Gibt den Datenverkehr an, der Ihre Instance erreichen kann. Wählen Sie die Datenverkehrquelle aus der Liste aus. Wenn Sie Custom (Benutzerdefiniert) auswählen, geben Sie eine einzelne IP-Adresse oder einen IP-Adressbereich CIDR-Notation an (z. B. 203.0.113.5/32).

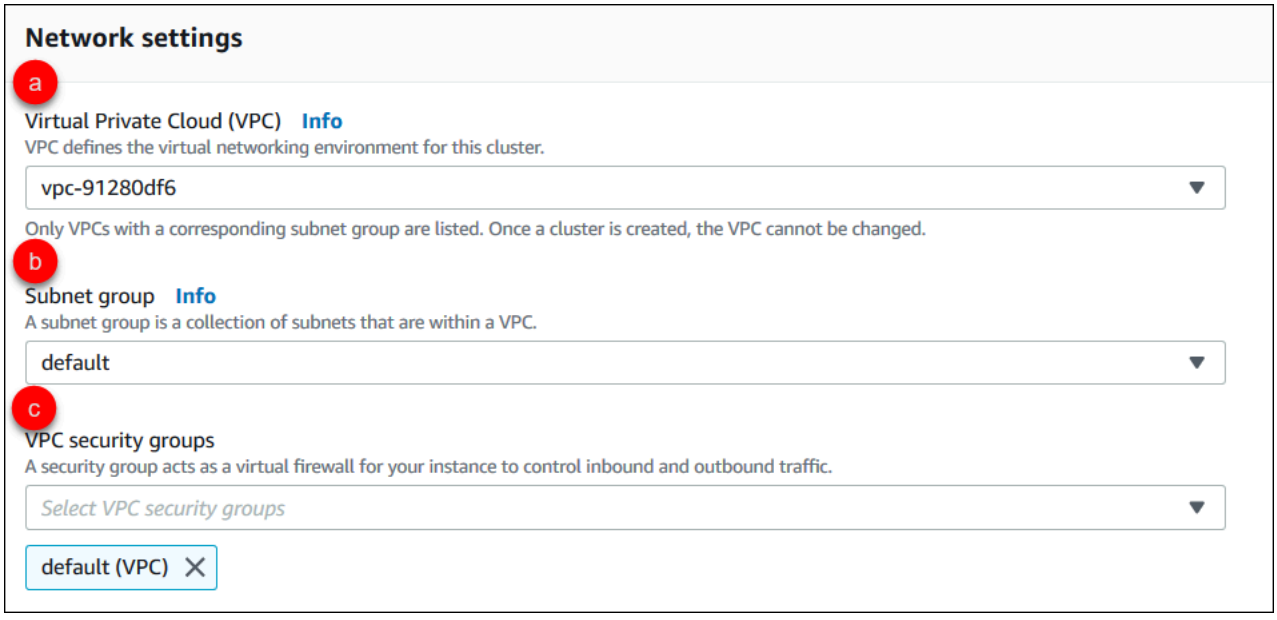
   v. Beschreibung – Geben Sie eine Beschreibung für diese Regel ein.

   vi. Wenn Sie die Regel fertig erstellt haben, wählen Sie Save (Speichern).

Erstellen eines Clusters: zusätzliche Konfigurationen

Wenn Sie die Standardeinstellungen für Ihren Cluster übernehmen möchten, können Sie die folgenden Schritte überspringen und Create cluster (Cluster erstellen) auswählen.

1. Vervollständigen Sie den Bereich Network settings (Netzwerkeinstellungen).



a. Virtual Private Cloud (VPC) – Wählen Sie in der Liste die Amazon VPC aus, in der Sie diesen Cluster starten möchten.

b. Subnetzgruppe – Wählen Sie in der Liste die Subnetzgruppe aus, die Sie für diesen Cluster verwenden möchten.

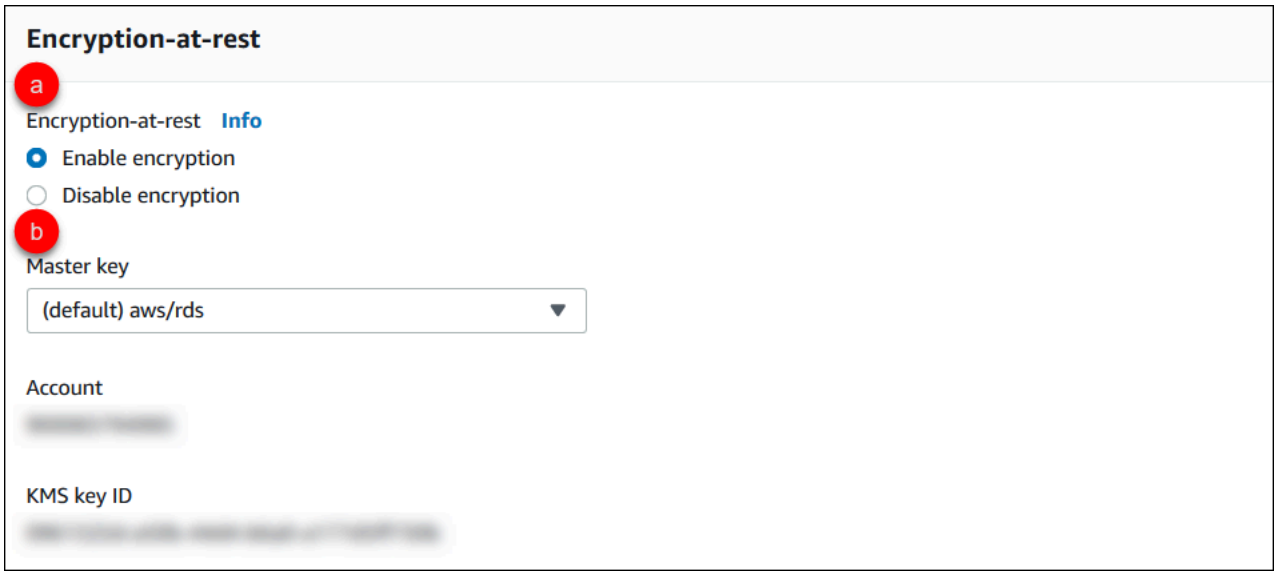
c. VPC-Sicherheitsgruppen – Wählen Sie in der Liste die VPC-Sicherheitsgruppe für diesen Cluster aus.

2. Vervollständigen Sie den Bereich Cluster options (Cluster-Optionen).

a. Datenbankport – Verwenden Sie die Aufwärts- und Abwärtspfeile, um den TCP/IP-Port festzulegen, den Anwendungen für die Verbindung mit Ihrer Instance verwenden.

b. Cluster-Parametergruppe – Wählen Sie in der Liste der Parametergruppen die Cluster-Parametergruppe für diesen Cluster aus.

3. Vervollständigen Sie den Bereich Encryption (Verschlüsselung).



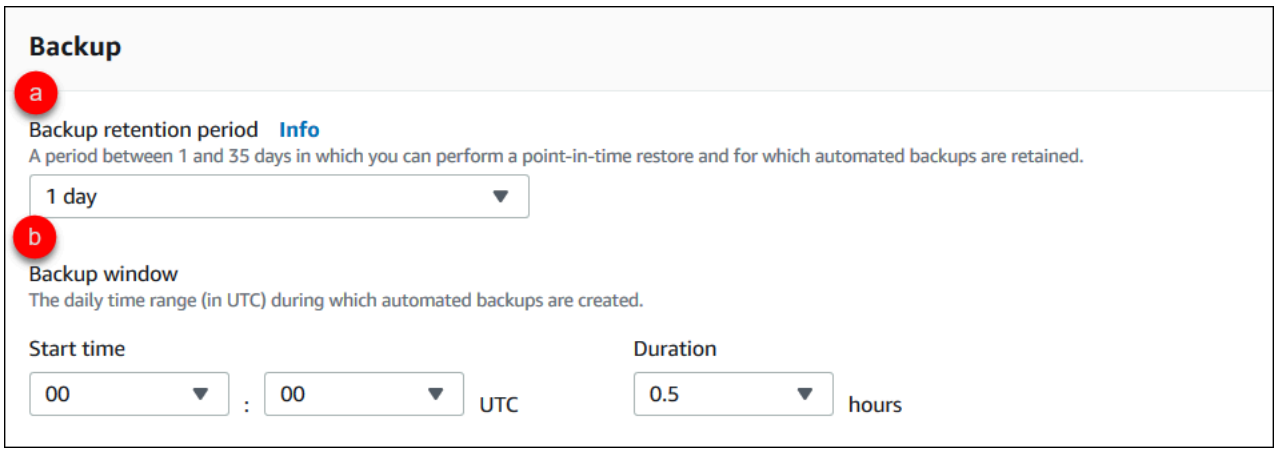
a. Encryption-at-rest – Wählen Sie eine der folgenden Optionen aus:

   - Verschlüsselung aktivieren – Standard. Alle Daten im Ruhezustand werden verschlüsselt. Wenn Sie sich für die Verschlüsselung Ihrer Daten entscheiden, können Sie diese Aktion nicht rückgängig machen.
   - Verschlüsselung deaktivieren – Ihre Daten sind nicht verschlüsselt.

b. Masterschlüssel – Dieser Schlüssel ist nur verfügbar, wenn Sie Ihre Daten verschlüsseln. Wählen Sie aus der Liste den Schlüssel aus, den Sie für die Verschlüsselung der Daten in diesem Cluster verwenden möchten. Der Standardwert ist `(default) aws/rds`.

   Wenn Sie Enter a key ARN (Einen Schlüssel-ARN eingeben) auswählen, müssen Sie einen Amazon-Ressourcenname (ARN) für den Schlüssel eingeben.

4. Vervollständigen Sie den Bereich Backup.



a. Aufbewahrungszeitraum für Backups – Wählen Sie in der Liste die Anzahl der Tage aus, für die automatische Backups dieses Clusters aufbewahrt werden sollen, bevor sie gelöscht werden.

b. Backup-Fenster – Legen Sie die tägliche Zeit und Dauer fest, in der Amazon DocumentDB Backups dieses Clusters erstellen soll.

   i. Startzeit – Wählen Sie in der ersten Liste die Startzeitstunde (UTC) zum Starten Ihrer automatischen Backups aus. Wählen Sie in der zweiten Liste die Minute für den Beginn der automatischen Backups aus.

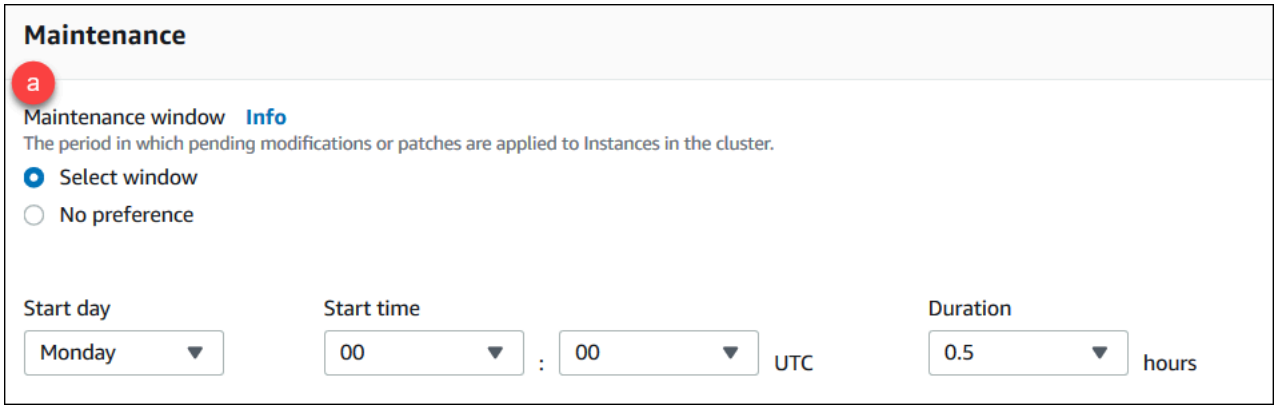
   ii. Dauer – Wählen Sie in der Liste die Anzahl der Stunden aus, die dem Erstellen automatischer Backups zugewiesen werden sollen.

5. Füllen Sie den Bereich Protokollexporte aus, indem Sie die Arten von Protokollen auswählen, die Sie in CloudWatch Protokolle exportieren möchten.

- Prüfungsprotokolle – Wählen Sie diese Option aus, um den Export von Prüfungsprotokollen nach Amazon CloudWatch Logs zu aktivieren. Wenn Sie Audit logs (Auditprotokolle), auswählen, müssen Sie `audit_logs` in der benutzerdefinierten Parametergruppe des Clusters aktivieren. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.
- Profiler-Protokolle – Wählen Sie diese Option aus, um den Export von Operationsprofiler-Protokollen nach Amazon CloudWatch Logs zu aktivieren. Wenn Sie Profiler logs (Profiler-Protokolle) auswählen, müssen Sie auch die folgenden Parameter in der benutzerdefinierten Parametergruppe des Clusters ändern:
  - `profiler`– Setzen Sie auf `enabled`.
  - `profiler_threshold_ms`– Setzen Sie auf einen Wert`[0-INT_MAX]`, um den Schwellenwert für Profilerstellungsvorgänge festzulegen.
  - `profiler_sampling_rate`– Legen Sie auf einen Wert fest`[0.0-1.0]`, um den Prozentsatz der langsamen Operationen für das Profil festzulegen.

  Weitere Informationen finden Sie unter Profilierung von Amazon DocumentDB-Vorgängen.

6. Schließen Sie den Bereich Maintenance (Wartung) ab.



- Wählen Sie eine der folgenden Optionen aus:
  - Fenster auswählen – Sie können den Wochentag, die UTC-Startzeit und die Dauer für Amazon DocumentDB angeben, um Wartungsarbeiten an Ihrem Cluster durchzuführen.

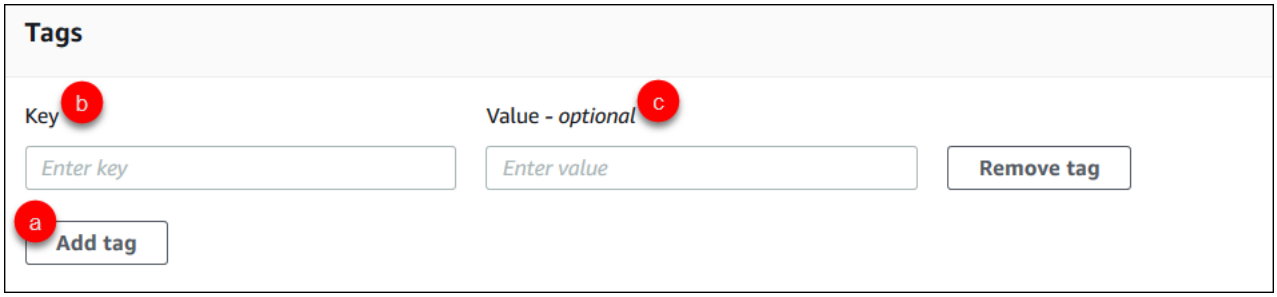
a. Starttag – Wählen Sie in der Liste den Wochentag aus, an dem die Cluster-Wartung gestartet werden soll.

b. Startzeit – Wählen Sie in den Listen die Stunde und Minute (UTC) aus, um mit der Wartung zu beginnen.

c. Dauer – Wählen Sie in der Liste aus, wie viel Zeit für die Cluster-Wartung zugewiesen werden soll. Wenn die Wartung nicht innerhalb der angegebenen Zeit abgeschlossen werden kann, wird der Wartungsprozess über die angegebene Zeit hinaus bis zum Abschluss fortgesetzt.

- Keine Präferenz – Amazon DocumentDB wählt den Wochentag, die Startzeit und die Dauer für die Durchführung von Wartungsarbeiten aus.

7. Wenn Sie diesem Cluster ein oder mehrere Tags hinzufügen möchten, stellen Sie den Bereich Tags fertig.



Wiederholen Sie die folgenden Schritte für alle Tags, die Sie dem Cluster hinzufügen möchten. Möglicherweise haben Sie bis zu 10 auf einem Cluster.

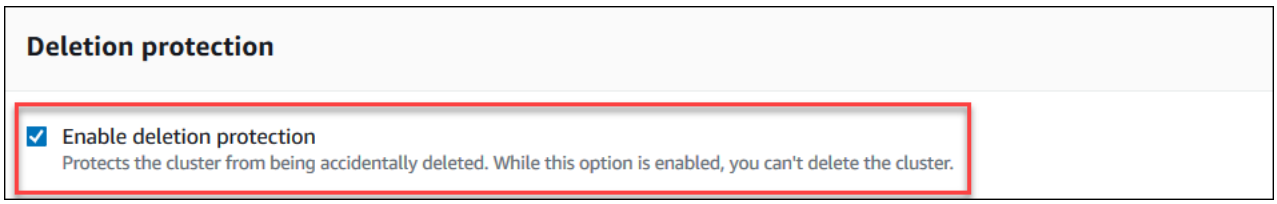
a. Wählen Sie Tags hinzufügen aus.

b. Geben Sie den Schlüssel des Tags ein.

c. Optional können Sie auch den Wert des Tags eingeben.

Klicken Sie zum Entfernen eines Tags auf Tag entfernen.

8. Der Löschschutz ist standardmäßig aktiviert, wenn Sie ein Cluster mit der Konsole erstellen. Deaktivieren Sie zum Ausschalten des Löschschutzes die Option Löschschutz aktivieren. Wenn diese Option aktiviert ist, verhindert der Löschschutz das Löschen eines Clusters. Wenn Sie einen löschgeschützten Cluster löschen möchten, müssen Sie zuerst den Cluster-Löschschutz deaktivieren.



Weitere Informationen über den Löschschutz finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

9. Wählen Sie zum Erstellen des Clusters Cluster erstellen aus. Wählen Sie andernfalls Abbrechen.

Erstellen eines Clusters mit der AWS CLI

In den folgenden Verfahren wird beschrieben, wie Sie mithilfe der einen Amazon AWS CLI - DocumentDB-Cluster starten und ein Amazon DocumentDB-Replikat erstellen. Amazon DocumentDB

Parameter

- **`--db-cluster-identifier`**—Erforderlich. Eine Zeichenfolge in Kleinbuchstaben, die diesen Cluster identifiziert.

  Einschränkungen bei der Benennung von Clustern:

  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster (über Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

- **`--engine`**—Erforderlich. Der Wert muss **`docdb`** sein.
- **`--deletion-protection | --no-deletion-protection`**—Optional. Der aktivierte Löschschutz verhindert, dass ein Cluster gelöscht wird. Wenn Sie die verwenden AWS CLI, ist der Löschschutz standardmäßig deaktiviert.

  Weitere Informationen über den Löschschutz finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

- **`--storage-type standard | iopt1`**—Optional. Standard: **`standard`**. Die Speicherkonfiguration des Clusters. Gültige Werte sind `standard` (Standard) oder `iopt1` (E/A-optimiert).

- **`--master-username`**—Erforderlich. Der Benutzername für die Authentifizierung des Benutzers.

  Namenseinschränkungen für Benutzer:

  - Die Länge beträgt [1 bis 63] alphanumerische Zeichen.
  - Muss mit einem Buchstaben beginnen.
  - Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

- **`--master-user-password`**—Erforderlich. Das Passwort für die Authentifizierung des Benutzers.

  Einschränkungen für Hauptpasswort:

  - Länge beträgt [8–100] druckbare ASCII-Zeichen.
  - Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
    - **`/`** (Schrägstrich)
    - **`"`** (doppeltes Anführungszeichen)
    - **`@`** ('At'-Symbol)

Weitere Parameter finden Sie unter CreateDBCluster.

So starten Sie einen Amazon DocumentDB-Cluster mit der AWS CLI

Um einen Amazon DocumentDB-Cluster zu erstellen, rufen Sie die auf`create-db-cluster` AWS CLI. Mit dem folgenden AWS CLI Befehl wird ein Amazon DocumentDB-Cluster mit dem Namen `sample-cluster` mit aktiviertem Löschschutz erstellt. Weitere Informationen zum Löschschutz finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Außerdem `--engine-version` ist ein optionaler Parameter, der standardmäßig die neueste Engine-Hauptversion verwendet. Die aktuelle Engine-Hauptversion ist 4.0.0. Wenn neue Engine-Hauptversionen veröffentlicht werden, `--engine-version` wird die Standard-Engine-Version für aktualisiert, um die zuletzt verwendete Engine-Hauptversion widerzuspiegeln. Daher empfehlen wir für Produktions-Workloads und insbesondere solche, die von Skripting, Automatisierung oder AWS CloudFormation Vorlagen abhängig sind, die explizite Angabe der `--engine-version` für die gewünschte Hauptversion.

> **Note**
>
> Wenn `db-subnet-group-name` oder nicht angegeben `vpc-security-group-id` ist, verwendet Amazon DocumentDB die Standardsubnetzgruppe und die Amazon-VPC-Sicherheitsgruppe für die angegebene Region.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 4.0.0 \
      --deletion-protection \
      --master-username masteruser \
      --master-user-password password
```

Für Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
      --engine-version 4.0.0 ^
      --deletion-protection ^
      --master-username masteruser ^
      --master-user-password password
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "4.0.0",
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
```

```
        "DBClusterIdentifier": "sample-cluster",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "sample-cluster.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "sample-cluster.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

Die Erstellung des Clusters dauert mehrere Minuten. Sie können die AWS Management Console oder verwenden AWS CLI , um den Status Ihres Clusters zu überwachen. Weitere Informationen finden Sie unter Überwachung des Status eines Amazon DocumentDB-Clusters.

> **Important**
>
> Wenn Sie die verwenden, AWS CLI um einen Amazon DocumentDB-Cluster zu erstellen, werden keine Instances erstellt. Daher müssen Sie explizit eine primäre Instance und alle benötigten Replikat-Instances anlegen. Sie können entweder die -Konsole oder verwenden AWS CLI , um die Instances zu erstellen. Weitere Informationen finden Sie unter Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster.

Weitere Informationen finden Sie unter `CreateDBCluster` in der Amazon DocumentDB-API-Referenz.

Beschreiben von Amazon DocumentDB-Clustern

Sie können entweder die Amazon DocumentDB-Managementkonsole oder die verwenden, AWS CLI um Details wie Verbindungsendpunkte, Sicherheitsgruppen, VPCs und Parametergruppen zu Ihren Amazon DocumentDB-Clustern anzuzeigen.

Weitere Informationen finden Sie hier:

- Überwachung des Status eines Amazon DocumentDB-Clusters
- Suchen der Endpunkte eines Clusters

Using the AWS Management Console

Gehen Sie wie folgt vor, um die Details eines angegebenen Amazon DocumentDB-Clusters mithilfe der Konsole anzuzeigen.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, dessen Details Sie sehen möchten. Die Informationen über den Cluster sind in die folgenden Gruppierungen unterteilt:

   - Zusammenfassung – Allgemeine Informationen zum Cluster, einschließlich der Engine-Version, des Cluster-Status, der ausstehenden Wartung und des Status seiner Parametergruppe.
   - Konnektivität und Sicherheit – Im Abschnitt Verbinden werden Verbindungsendpunkte aufgeführt, um über die mongo-Shell oder mit einer Anwendung eine Verbindung zu diesem Cluster herzustellen. Im Abschnitt Sicherheitsgruppen werden die

Sicherheitsgruppen aufgeführt, die diesem Cluster zugeordnet sind, sowie ihre VPC-ID und Beschreibungen.

- Konfiguration – Im Abschnitt Cluster-Details werden Details zum Cluster aufgeführt, einschließlich des Amazon-Ressourcennamens (ARN), des Endpunkts und der Parametergruppe des Clusters. Außerdem werden die Sicherungsinformationen, Wartungsdetails sowie Sicherheits- und Netzwerkeinstellungen des Clusters aufgelistet. Im Abschnitt Cluster-Instances werden die Instances aufgeführt, die zu diesem Cluster gehören, mit dem Status der einzelnen Instance-Rollen und Cluster-Parametergruppen.
- Überwachung – Die Amazon- CloudWatch Logs-Metriken für diesen Cluster. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.
- Ereignisse und Tags – Im Abschnitt Aktuelle Ereignisse werden die letzten Ereignisse für diesen Cluster aufgeführt. Amazon DocumentDB zeichnet Ereignisse auf, die sich auf Ihre Cluster, Instances, Snapshots, Sicherheitsgruppen und Cluster-Parametergruppen beziehen. Zu diesen Informationen gehören Datum, Uhrzeit und Nachricht, die jedem Ereignis zugeordnet sind. Der Abschnitt Tags listet die an diesen Cluster angehängten Tags auf.

Using the AWS CLI

Um die Details Ihrer Amazon DocumentDB-Cluster mithilfe der anzuzeigen AWS CLI, verwenden Sie den Befehl , `describe-db-clusters` wie in den folgenden Beispielen gezeigt. Weitere Informationen finden Sie unter `DescribeDBClusters` in der API-Referenz für Amazon DocumentDB Resource Management.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen wie Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS geteilt wird. Der `filterName=engine,Values=docdb` Filterparameter gibt nur Amazon DocumentDB-Cluster zurück.

Example

Beispiel 1: Auflisten aller Amazon DocumentDB-Cluster

Der folgende AWS CLI Code listet die Details für alle Amazon DocumentDB-Cluster in einer Region auf.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusters": [
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-1",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-2",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-3",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
```

```
            ...
        }
    ]
}
```

Example

Beispiel 2: Auflisten aller Details für einen angegebenen Amazon DocumentDB-Cluster

Der folgende AWS CLI Code listet die Details für den Cluster auf`sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
   --filter Name=engine,Values=docdb \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb describe-db-clusters ^
   --filter Name=engine,Values=docdb ^
   --db-cluster-identifier sample-cluster
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusters": [
        {
            "AllocatedStorage": 1,
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1a",
                "us-east-1d"
            ],
            "BackupRetentionPeriod": 2,
            "DBClusterIdentifier": "sample-cluster",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            "EarliestRestorableTime": "2023-11-07T22:34:08.148000+00:00",
            "Endpoint": "sample-cluster.node.us-east-1.amazon.com",
            "ReaderEndpoint": "sample-cluster.node.us-east-1.amazon.com",
            "MultiAZ": false,
```

```
            "Engine": "docdb",
            "EngineVersion": "5.0.0",
            "LatestRestorableTime": "2023-11-10T07:21:16.772000+00:00",
            "Port": 27017,
            "MasterUsername": "chimeraAdmin",
            "PreferredBackupWindow": "22:22-22:52",
            "PreferredMaintenanceWindow": "sun:03:01-sun:03:31",
            "ReadReplicaIdentifiers": [],
            "DBClusterMembers": [
                {
                    "DBInstanceIdentifier": "sample-instance-1",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "sample-instance-2",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },

            ],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-9084c2ec",
                    "Status": "active"
                }
            ],
            "HostedZoneId": "Z06853723JYKYBXTJ49RB",
            "StorageEncrypted": false,
            "DbClusterResourceId": "cluster-T4LGLANHVAPGQYYULWUDKLVQL4",
            "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster",
            "AssociatedRoles": [],
            "IAMDatabaseAuthenticationEnabled": false,
            "ClusterCreateTime": "2023-11-06T18:05:41.568000+00:00",
            "EngineMode": "provisioned",
            "DeletionProtection": false,
            "HttpEndpointEnabled": false,
            "CopyTagsToSnapshot": false,
            "CrossAccountClone": false,
            "DomainMemberships": [],
            "TagList": [],
```

```
            "StorageType": "iopt1",
            "AutoMinorVersionUpgrade": false,
            "NetworkType": "IPV4",
            "IOOptimizedNextAllowedModificationTime":
 "2023-12-07T18:05:41.580000+00:00"
        }
    ]
}
```

Example

Beispiel 3: Auflisten bestimmter Details für einen Amazon DocumentDB-Cluster

Um eine Teilmenge der Cluster-Details mithilfe der aufzulisten AWS CLI, fügen Sie eine hinzu, `--query` die angibt, welche Cluster-Mitglieder die `describe-db-clusters` Operation auflisten soll. Der `--db-cluster-identifier`-Parameter ist der Bezeichner für den jeweiligen Cluster, dessen Details Sie anzeigen möchten. Weitere Informationen zu Abfragen finden Sie unter Filtern der Ausgabe mit der `--query` Option im AWS Command Line Interface -Benutzerhandbuch.

Im folgenden Beispiel werden die Instances in einem Amazon DocumentDB-Cluster aufgelistet.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --filter Name=engine,Values=docdb \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --filter Name=engine,Values=docdb ^
    --db-cluster-identifier sample-cluster ^
    --query 'DBClusters[*].[DBClusterMembers]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
```

```
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ]
    ]
]
```

Ändern eines Amazon DocumentDB-Clusters

Nur Cluster, die sich im Status available befinden, können geändert werden. Ein angehaltener Cluster kann nicht geändert werden. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis der Cluster verfügbar wird, und nehmen Sie dann die gewünschten Änderungen vor. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

Using the AWS Management Console

Gehen Sie wie folgt vor, um einen bestimmten Amazon DocumentDB-Cluster mithilfe der Konsole zu ändern.

So ändern Sie einen Amazon DocumentDB-Cluster

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Geben Sie den Cluster an, den Sie ändern möchten, indem Sie die Schaltfläche links neben dem Namen des Clusters auswählen.
4. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
5. Nehmen Sie im Bereich Modify Cluster: <cluster-name> (Cluster ändern: <cluster-name>) die gewünschten Änderungen vor. Sie können in den folgenden Bereichen Änderungen durchführen:

   - Cluster-Spezifikationen – Der Name, die Sicherheitsgruppen und das Passwort des Clusters.
   - Cluster-Speicherkonfiguration – Der Datenspeichermodus des Clusters. Wählen Sie zwischen Standard- und E/A-optimierter Konfiguration.
   - Cluster-Optionen – Der Port und die Parametergruppe des Clusters.
   - Backup – Der Aufbewahrungszeitraum für Backups und das Backup-Fenster des Clusters.
   - Protokollexporte – Aktivieren oder deaktivieren Sie den Export von Audit- oder Profilerprotokollen.
   - Wartung – Legen Sie das Wartungsfenster des Clusters fest.
   - Löschschutz – Aktivieren oder deaktivieren Sie den Löschschutz auf dem Cluster. Der Löschschutz ist standardmäßig aktiviert.
6. Wenn Sie fertig sind, wählen Sie Continue (Weiter) aus, um eine Zusammenfassung Ihrer Änderungen anzuzeigen.
7. Wenn Sie mit den Änderungen zufrieden sind, können Sie Modify cluster (Cluster ändern) auswählen, um den Cluster zu ändern. Alternativ können Sie Back (Zurück) oder Cancel (Abbrechen) wählen, um die Änderungen zu bearbeiten oder abzubrechen.

Es dauert einige Minuten, bis Ihre Änderungen übernommen werden. Sie können den Cluster nur verwenden, wenn sein Status available ist. Sie können den Status des Clusters über die Konsole oder AWS CLI überwachen. Weitere Informationen finden Sie unter Überwachung des Status eines Amazon DocumentDB-Clusters.

Using the AWS CLI

Verwenden Sie die `modify-db-cluster`-Operation, um den angegebenen Cluster mit der AWS CLI zu ändern. Weitere Informationen finden Sie unter `ModifyDBCluster` in der Amazon DocumentDB-API-Referenz.

Parameter

- **`--db-cluster-identifier`**—Erforderlich. Die Kennung des Amazon DocumentDB-Clusters, den Sie ändern möchten.
- **`--backup-retention-period`**—Optional. Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Gültige Werte sind 1–35.
- **`--storage-type`**—Optional. Die Speicherkonfiguration des Clusters. Gültige Werte sind `standard` (Standard) oder `iopt1` (E/A-optimiert).
- **`--db-cluster-parameter-group-name`**—Optional. Der Name der Cluster-Parametergruppe, die für den Cluster verwendet werden soll.
- **`--master-user-password`**—Optional. Das neue Passwort für den Masterbenutzer der Datenbank.

  Passwortbeschränkungen:

  - Die Länge beträgt [8–100] druckbare ASCII-Zeichen.
  - Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
    - **`/`** (Schrägstrich)
    - **`"`** (doppeltes Anführungszeichen)
    - **`@`** ('At'-Symbol)
- **`--new-db-cluster-identifier`**—Optional. Die neue Cluster-ID für den Cluster beim Umbenennen eines Clusters. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

  Benennungseinschränkungen:

  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Amazon Neptune und Amazon DocumentDB pro und AWS-KontoRegion eindeutig sein.
- **`--preferred-backup-window`**—Optional. Der tägliche Zeitbereich, in dem automatisierte Backups erstellt werden, in Universal Coordinated Time (UTC).
  - Format: `hh24:mm-hh24:mm`

- **`--preferred-maintenance-window`**—Optional. Der wöchentliche Zeitbereich, in dem Systemwartungen stattfinden können, in UTC.
  - Format: `ddd:hh24:mm-ddd:hh24:mm`
  - Gültige Tage: `Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri` und `Sat`.
- **`--deletion-protection`** oder **`--no-deletion-protection`**—Optional. Gibt an, ob der Löschschutz für diesen Cluster aktiviert werden soll. Der Löschschutz verhindert, dass ein Cluster versehentlich gelöscht wird, solange der Cluster-Löschschutz nicht deaktiviert wurde. Weitere Informationen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.
- **`--apply-immediately`** oder **`--no-apply-immediately`**– Verwenden Sie `--apply-immediately`, um die Änderung sofort vorzunehmen. Verwenden Sie `--no-apply-immediately`, um die Änderung während des nächsten Wartungsfensters Ihres Clusters vorzunehmen.

Example

Der folgende Code ändert die Aufbewahrungsfrist für das Backup für den Cluster `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --apply-immediately \
       --backup-retention-period 7
```

Für Windows:

```
aws docdb modify-db-cluster ^
       --db-cluster-identifier sample-cluster ^
       --apply-immediately ^
       --backup-retention-period 7
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "BackupRetentionPeriod": 7,
        "DbClusterResourceId": "cluster-VDP53QEWST7YHM36TTXOPJT5YE",
        "Status": "available",
```

```
        "DBClusterMembers": [
            {
                "PromotionTier": 1,
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster-instance",
                "IsClusterWriter": true
            }
        ],
        "ReadReplicaIdentifiers": [],
        "AvailabilityZones": [
            "us-east-1b",
            "us-east-1c",
            "us-east-1a"
        ],
        "ReaderEndpoint": "sample-cluster.cluster-ro-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "PreferredMaintenanceWindow": "sat:09:51-sat:10:21",
        "EarliestRestorableTime": "2018-06-17T00:06:19.374Z",
        "StorageEncrypted": false,
        "MultiAZ": false,
        "AssociatedRoles": [],
        "MasterUsername": "<your-master-user-name>",
        "DBClusterIdentifier": "sample-cluster",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "HostedZoneId": "Z2SUY0A1719RZT",
        "LatestRestorableTime": "2018-06-18T21:17:05.737Z",
        "AllocatedStorage": 1,
        "Port": 27017,
        "Engine": "docdb",
        "DBClusterParameterGroup": "default.docdb3.4",
        "Endpoint": "sample-cluster.cluster-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
        "DBSubnetGroup": "default",
        "PreferredBackupWindow": "00:00-00:30",
        "EngineVersion": "3.4",
        "ClusterCreateTime": "2018-06-06T19:25:47.991Z",
        "IAMDatabaseAuthenticationEnabled": false
    }
```

```
}
```

Es dauert einige Minuten, bis Ihre Änderungen übernommen werden. Sie können den Cluster nur verwenden, wenn sein Status available ist. Sie können den Status des Clusters über die Konsole oder AWS CLI überwachen. Weitere Informationen finden Sie unter Überwachung des Status eines Amazon DocumentDB-Clusters.

Ermittlung ausstehender Wartungsarbeiten

Sie können feststellen, ob Sie über die neueste Amazon DocumentDB-Engine-Version verfügen, indem Sie feststellen, ob die Cluster-Wartung aussteht.

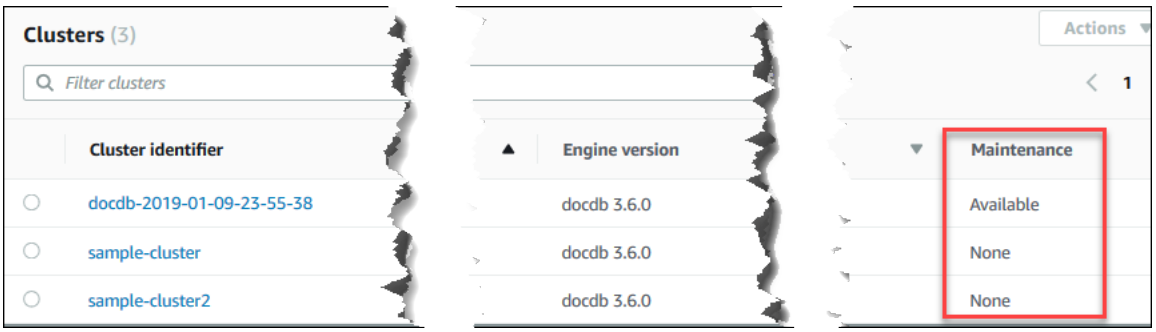
Using the AWS Management Console

Sie können die verwenden AWS Management Console , um festzustellen, ob für einen Cluster die Wartung aussteht.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Überprüfen Sie die Spalte Maintenance (Wartung), um festzustellen, ob ein Cluster eine ausstehende Wartung hat.



None (Keine) bedeutet, dass auf dem Cluster die neueste Engine-Version ausgeführt wird. Available (Verfügbar) bedeutet, dass der Cluster noch nicht gewartet wurde. Somit ist ein Engine-Upgrade erforderlich.

4. Wenn Ihr Cluster noch nicht gewartet wurde, fahren Sie mit den Schritten unter Durchführen eines Patch-Updates für die Engine-Version eines Clusters fort.

Using the AWS CLI

Sie können die verwenden AWS CLI , um festzustellen, ob ein Cluster die neueste Engine-Version hat, indem Sie die -`describe-pending-maintenance-actions`Operation mit den folgenden Parametern verwenden.

Parameter

- **`--resource-identifier`**—Optional. Der ARN der Ressource (Cluster). Wenn dieser Parameter weggelassen wird, werden ausstehende Wartungsaktionen für alle Cluster aufgelistet.
- **`--region`**—Optional. Die AWS-Region, in der Sie diese Operation durchführen möchten (z. B. `us-east-1`).

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-actions \
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster \
   --region us-east-1
```

Für Windows:

```
aws docdb describe-pending-maintenance-actions ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --region us-east-1
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:123456789012:cluster:sample-cluster",
            "PendingMaintenanceActionDetails": [
                {
                    "Description": "New feature",
                    "Action": "db-upgrade",
                    "ForcedApplyDate": "2019-02-25T21:46:00Z",
                    "AutoAppliedAfterDate": "2019-02-25T07:41:00Z",
                    "CurrentApplyDate": "2019-02-25T07:41:00Z"
                }
            ]
        }
    ]
}
```

Wenn Ihr Cluster noch nicht gewartet wurde, fahren Sie mit den Schritten unter Durchführen eines Patch-Updates für die Engine-Version eines Clusters fort.

Durchführen eines Patch-Updates für die Engine-Version eines Clusters

In diesem Abschnitt wird erläutert, wie Sie ein Patch-Update mithilfe der AWS Management Console oder der bereitstellen AWS CLI. Ein Patch-Update ist ein Update innerhalb derselben Engine-Version (z. B. das Aktualisieren einer 3.6-Engine-Version auf eine neuere 3.6-Engine-Version). Sie können sie sofort oder während des nächsten Wartungsfensters Ihres Clusters aktualisieren. Informationen dazu, ob Ihre Engine ein Update benötigt, finden Sie unter Ermittlung ausstehender Wartungsarbeiten. Bitte beachten Sie, dass es bei der Anwendung des Updates zu Ausfallzeiten in Ihrem Cluster kommt.

> **Note**
>
> Wenn Sie versuchen, von einer Engine-Hauptversion auf eine andere zu aktualisieren, z. B. 3.6 auf 5.0, finden Sie weitere Informationen unter Direktes Upgrade der Hauptversion von Amazon DocumentDB oder Aktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service. Ein direktes Hauptversions-Upgrade unterstützt nur docdb 5.0 als Ziel-Engine-Version.

Es gibt zwei Konfigurationsanforderungen, um die neuesten Patch-Updates für die Engine-Version eines Clusters zu erhalten:

- Der Cluster-Status muss available (verfügbar) lauten.
- Der Cluster muss eine ältere Engine-Version ausführen.

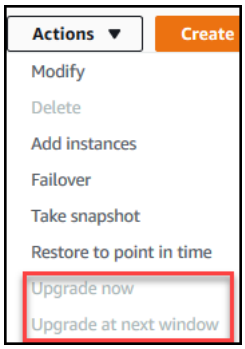
Using the AWS Management Console

Das folgende Verfahren wendet Patch-Updates auf die Engine-Version Ihres Clusters mithilfe der Konsole an. Sie haben die Möglichkeit, sofort oder während des nächsten Wartungsfensters Ihres Clusters zu aktualisieren.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Cluster aus, den Sie upgraden möchten. Der Cluster-Status muss available (verfügbar) lauten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Actions (Aktionen) eine der folgenden Optionen aus. Diese Menüoptionen sind nur auswählbar, wenn der ausgewählte Cluster nicht die neueste Engine-Version ausführt.



   - Jetzt aktualisieren – Initiiert den Upgrade-Prozess sofort. Ihr Cluster ist eine zeitlang offline, während er auf die neueste Engine-Version aktualisiert wird.

- Upgrade im nächsten Fenster – Initiiert den Upgrade-Prozess während des nächsten Wartungsfensters des Clusters. Ihr Cluster ist eine zeitlang offline, während er auf die neueste Engine-Version aktualisiert wird.

4. Wenn sich das Bestätigungsfenster öffnet, wählen Sie eine der folgenden Optionen aus:
   - Upgrade – Um Ihren Cluster auf die neueste Engine-Version gemäß dem im vorherigen Schritt ausgewählten Zeitplan zu aktualisieren.
   - Abbrechen – Um das Engine-Upgrade des Clusters abzubrechen und mit der aktuellen Engine-Version des Clusters fortzufahren.

Using the AWS CLI

Sie können Patch-Updates auf Ihren Cluster anwenden, indem Sie die AWS CLI und die `apply-pending-maintenance-action`Operation mit den folgenden Parametern verwenden.

Parameter

- **`--resource-identifier`**—Erforderlich. Der ARN des Amazon DocumentDB-Clusters, den Sie aktualisieren möchten.
- **`--apply-action`**—Erforderlich. Die folgenden Werte sind zulässig. Verwenden Sie `db-upgrade`, um Ihre Cluster-Engine-Version upzugraden.
  - **`db-upgrade`**
  - **`system-update`**
- **`--opt-in-type`**—Erforderlich. Die folgenden Werte sind zulässig.
  - `immediate`– Wenden Sie die Wartungsaktion sofort an.
  - `next-maintenance`– Wenden Sie die Wartungsaktion während des nächsten Wartungsfensters an.
  - `undo-opt-in`– Bricht alle vorhandenen `next-maintenance` Opt-In-Anforderungen ab.

Example

Der folgende Beispiel-Patch aktualisiert die Engine-Version von `sample-cluster` auf Version 4.0.0.

Für Linux, macOS oder Unix:

```
aws docdb apply-pending-maintenance-action \
```

```
   --resource-identifier arn:aws:rds:us-east-1:123456789012\:cluster:sample-cluster
 \
   --apply-action db-upgrade \
   --opt-in-type immediate
```

Für Windows:

```
aws docdb apply-pending-maintenance-action ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --apply-action db-upgrade ^
   --opt-in-type immediate
```

Die Ausgabe dieser Operation sieht folgendermaßen aus.

```
{
    "ResourcePendingMaintenanceActions": {
        "ResourceIdentifier": "arn:aws:rds:us-
east-1:444455556666:cluster:docdb-2019-01-09-23-55-38",
        "PendingMaintenanceActionDetails": [
            {
                "CurrentApplyDate": "2019-02-20T20:57:06.904Z",
                "Description": "Bug fixes",
                "ForcedApplyDate": "2019-02-25T21:46:00Z",
                "OptInStatus": "immediate",
                "Action": "db-upgrade",
                "AutoAppliedAfterDate": "2019-02-25T07:41:00Z"
            }
        ]
    }
}
```

Stoppen und Starten eines Amazon DocumentDB-Clusters

Das Anhalten und Starten von Amazon DocumentDB-Clustern kann Ihnen helfen, die Kosten für Entwicklungs- und Testumgebungen zu verwalten. Anstatt Cluster und Instances bei jeder Verwendung von Amazon DocumentDB zu erstellen und zu löschen, können Sie alle Instances in Ihrem Cluster vorübergehend anhalten, wenn sie nicht benötigt werden. Sie können sie neu starten, wenn Sie Ihre Tests fortsetzen.

Themen

Übersicht über das Stoppen und Starten eines Clusters

In Zeiten, in denen Sie keinen Amazon DocumentDB-Cluster benötigen, können Sie alle Instances in diesem Cluster gleichzeitig stoppen. Bei Bedarf können Sie den Cluster jederzeit erneut starten. Durch das Starten und Stoppen werden die Einrichtungs- und Entfernungsvorgänge für Cluster erleichtert, die in der Entwicklung, für Tests oder ähnliche Aktivitäten verwendet werden und keine kontinuierliche Verfügbarkeit erfordern. Sie können einen Cluster mit der AWS Management Console oder der AWS CLI mit einer einzigen Aktion anhalten und starten, unabhängig davon, wie viele Instances sich im Cluster befinden.

Während Ihr Cluster gestoppt ist, bleibt des Cluster-Speichervolumen unverändert. Sie zahlen nur für Speicherung, manuelle Snapshots und automatischen Sicherungsspeicher innerhalb des angegebenen Aufbewahrungsfensters. Ihnen werden keine Instance-Stunden in Rechnung gestellt. Amazon DocumentDB startet Ihren Cluster nach sieben Tagen automatisch, damit er nicht hinter erforderlichen Wartungsupdates zurückfällt. Wenn Ihr Cluster nach sieben Tagen wieder startet, werden Ihnen die Instances im Cluster wieder in Rechnung gestellt. Während Ihr Cluster gestoppt ist, können Sie Ihr Speichervolumen nicht abfragen, da die Abfrage erfordert, dass sich Instances im Status „available (verfügbar)“ befinden.

Wenn ein Amazon DocumentDB-Cluster gestoppt wird, können weder der Cluster noch seine Instances in irgendeiner Weise geändert werden. Dies umfasst das Hinzufügen oder Entfernen von Instances oder das Löschen des Clusters.

Using the AWS Management Console

Das folgende Verfahren zeigt, wie Sie einen Cluster mit einer oder mehreren Instances im Status „verfügbar“ stoppen oder einen gestoppten Cluster wieder starten.

So stoppen oder starten Sie einen Amazon DocumentDB-Cluster

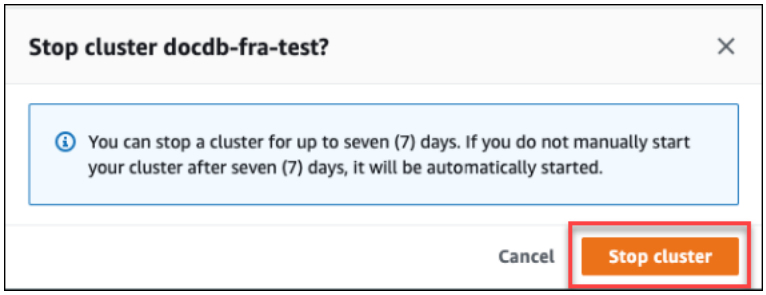
1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

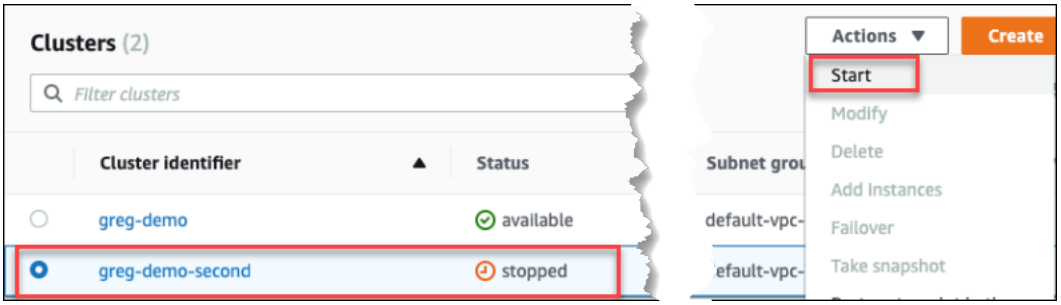
3. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Namen des Clusters aus, den Sie stoppen oder starten möchten.
4. Wählen Sie Aktionen aus und dann die Aktion, die Sie auf dem Cluster ausführen möchten.
   - Wenn Sie den Cluster stoppen möchten und der Cluster verfügbar ist:
     a. Wählen Sie Beenden aus.

Damit der Failover-Mechanismus nicht aktiviert wird, werden beim Stoppvorgang zuerst die Replikat-Instances und dann die primäre Instance gestoppt.

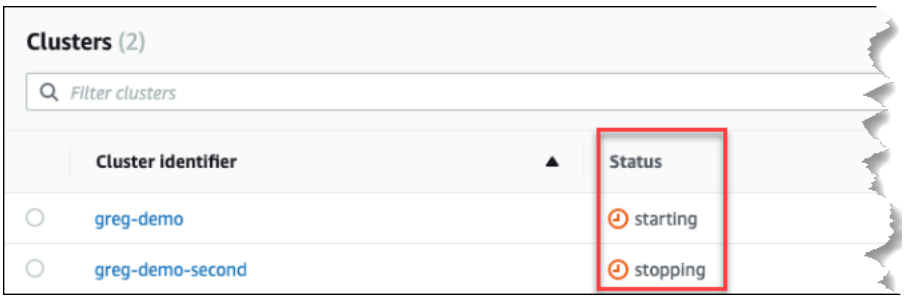
     b. Wenn Sie den Cluster stoppen möchten, wählen Sie im Bestätigungsdialogfeld Cluster stoppen aus und wenn der Cluster weiter ausgeführt werden soll, wählen Sie Abbrechen aus.



   - Wenn Sie einen gestoppten Cluster starten möchten, wählen Sie Starten aus.



5. Überwachen Sie den Status des Clusters und seiner Instances. Wenn Sie den Cluster gestartet haben, können Sie die Ausführung des Clusters fortsetzen, wenn der Cluster und dessen Instances verfügbar sind. Weitere Informationen finden Sie unter Ermitteln des Cluster-Status.



Using the AWS CLI

Die folgenden Codebeispiele zeigen, wie Sie einen Cluster mit einer oder mehreren Instances im Status „verfügbar“ stoppen oder einen gestoppten Cluster starten.

Um einen Cluster mit einer oder mehreren verfügbaren Instances mithilfe der zu stoppen AWS CLI, verwenden Sie die `-stop-db-cluster`Operation. Verwenden Sie die `start-db-cluster-`Operation, um einen gestoppten Cluster zu starten. Beide Operationen verwenden den `--db-cluster-identifier`-Parameter.

Parameter:

- **`--db-cluster-identifier`**—Erforderlich. Der Name des Clusters, der gestoppt oder gestartet werden soll.

Example – So stoppen Sie einen Cluster mit der AWS CLI

Der folgende Code stoppt den Cluster `sample-cluster`. Der Cluster muss eine oder mehrere Instances im verfügbaren Zustand haben.

Für Linux, macOS oder Unix:

```
aws docdb stop-db-cluster \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb stop-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Example – So starten Sie einen Cluster mit der AWS CLI

Der folgende Code startet den Cluster `sample-cluster`. Der Cluster muss jetzt gestoppt werden.

Für Linux, macOS oder Unix:

```
aws docdb start-db-cluster \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb start-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Operationen, die Sie auf einem gestoppten Cluster ausführen können

Während ein Amazon DocumentDB-Cluster gestoppt ist, können Sie eine point-in-time Wiederherstellung auf einen beliebigen Punkt innerhalb des von Ihnen angegebenen Zeitfensters für die Aufbewahrung automatisierter Backups durchführen. Weitere Informationen zum Durchführen einer point-in-time Wiederherstellung finden Sie unter Wiederherstellen auf einen bestimmten Zeitpunkt.

Sie können die Konfiguration eines Amazon DocumentDB-Clusters oder seiner Instances nicht ändern, während der Cluster gestoppt ist. Es ist auch nicht möglich, Instances zum Cluster hinzuzufügen oder daraus zu entfernen oder den Cluster zu löschen, wenn ihm noch Instances zugeordnet sind. Vor solchen administrativen Aktionen müssen Sie den Cluster starten.

Amazon DocumentDB wendet alle geplanten Wartungsarbeiten auf Ihren gestoppten Cluster erst an, nachdem er erneut gestartet wurde. Nach sieben Tagen startet Amazon DocumentDB automatisch

einen gestoppten Cluster, sodass er in Bezug auf seinen Wartungsstatus nicht zu weit zurückfällt. Sobald Ihr Cluster neu gestartet wird, werden Ihnen die Instances im Cluster wieder in Rechnung gestellt.

Während ein Cluster gestoppt wird, führt Amazon DocumentDB keine automatisierten Backups durch und verlängert auch nicht den Aufbewahrungszeitraum für Backups.

Löschen eines Amazon DocumentDB-Clusters

Sie können einen Amazon DocumentDB-Cluster mit der AWS Management Console oder der löschen AWS CLI. Um einen Cluster zu löschen, muss sich der Cluster im Status available (verfügbar) befinden und es dürfen keine Instances mit ihm verknüpft sein. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis der Cluster verfügbar wird, und löschen Sie dann den Cluster. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

Löschschutz

Zum Schutz Ihrer Cluster vor versehentlichem Löschen, können Sie den Löschschutz aktivieren. Wenn Sie mithilfe der Konsole einen Cluster erstellen, ist der Löschschutz standardmäßig aktiviert. Der Löschschutz ist jedoch standardmäßig deaktiviert, wenn Sie einen Cluster über die AWS CLI erstellen.

Amazon DocumentDB erzwingt Löschschutz für einen Cluster, unabhängig davon, ob Sie den Löschvorgang mit der Konsole oder der durchführen AWS CLI. Sie können einen Cluster nicht löschen, solange der Löschschutz aktiviert ist. Um einen Cluster mit aktiviertem Löschschutz zu löschen, müssen Sie zuerst den Cluster ändern und den Löschschutz deaktivieren.

Wenn Sie die Konsole mit aktiviertem Löschschutz für einen Cluster verwenden, können Sie die letzte Instance des Clusters nicht löschen, da dadurch auch der Cluster gelöscht werden würde. Sie können die letzte Instance eines mit Löschschutz geschützten Clusters mit der AWS CLI löschen. In diesen Fall bleibt der Cluster selbst bestehen und die Daten werden beibehalten. Sie können zum Zugriff auf die Daten neue Instances für den Cluster erstellen. Weitere Informationen zum Aktivieren und Deaktivieren des Löschschutzes siehe:

- Erstellen eines Amazon DocumentDB-Clusters
- Ändern eines Amazon DocumentDB-Clusters

Using the AWS Management Console

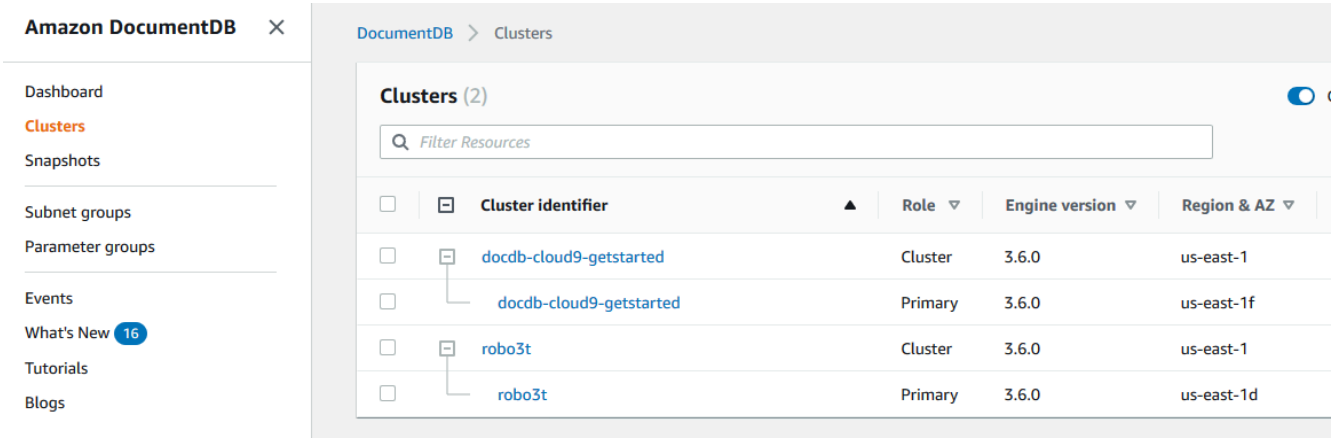
Um einen Cluster mit der zu löschen AWS Management Console, muss der Löschschutz deaktiviert sein.

So finden Sie heraus, ob der Löschschutz für einen Cluster aktiviert ist:

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Beachten Sie, dass im Navigationsfeld Cluster in der Spalte Cluster Identifier sowohl Cluster als auch Instances angezeigt werden. Instances sind unter Clustern aufgeführt, ähnlich wie im folgenden Screenshot.



4. Wählen Sie den Namen des Clusters und die Registerkarte Configuration (Konfiguration) aus. Suchen Sie im Abschnitt Cluster-Details die Option Deletion protection (Löschschutz). Wenn der Löschschutz aktiviert ist, bearbeiten Sie den Cluster, um den Löschschutz zu deaktivieren. Informationen über das Ändern eines Clusters finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

   Nachdem Sie den Löschschutz deaktiviert haben, können Sie den Cluster löschen.

So löschen Sie einen Cluster:

1. Klicken Sie im Navigationsbereich auf Cluster.
2. Sehen Sie in der Spalte Instances nach, ob der Cluster Instances hat. Bevor Sie einen Cluster löschen können, müssen Sie alle seine Instances löschen. Weitere Informationen finden Sie unter Löschen einer Amazon DocumentDB-Instance .
3. Abhängig davon, ob Ihr Cluster Instances hat oder nicht, führen Sie einen der folgenden Schritte aus:
   - Wenn der Cluster über keine Instances verfügt, wählen Sie die Schaltfläche links neben dem Cluster-Namen und dann Actions (Aktionen) aus. Wählen Sie im Dropdown-Menü Löschen aus. Füllen Sie das Dialogfeld <cluster-name> löschen aus und klicken Sie dann auf Löschen.
   - Wenn der Cluster eine oder mehrere Instances hat, gehen Sie wie folgt vor:
     a. Wählen Sie im Navigationsbereich Instances aus.
     b. Löschen Sie alle Instances des Clusters. Wenn Sie die letzte Instance löschen, wird der Cluster ebenfalls gelöscht. Weitere Informationen zum Löschen von Instances finden Sie unter Löschen einer Amazon DocumentDB-Instance .

Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Status des Clusters finden Sie unter Überwachung des Status eines Amazon DocumentDB-Clusters.

Using the AWS CLI

Sie können einen Cluster, dem Instances zugeordnet sind, nicht löschen. Um festzustellen, welche Instances mit dem Cluster verknüpft sind, führen Sie den Befehl `describe-db-clusters` aus und löschen Sie alle Instances des Clusters. Deaktivieren Sie dann bei Bedarf den Löschschutz auf Ihrem Cluster und löschen Sie schließlich den Cluster.

1. Löschen Sie zuerst alle Instances des Clusters.

   Um festzustellen, welche Instances Sie löschen müssen, führen Sie den folgenden Befehl aus.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].
[DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "sample-cluster",
        [
            "sample-instance-1",
            "sample-instance-2"
        ]
    ]
]
```

Wenn der Cluster, den Sie löschen möchten, Instances enthält, löschen Sie diese wie unten dargestellt.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-instance
```

2. Zweitens deaktivieren Sie den Löschschutz.

Wenn Sie die verwenden AWS CLI , um alle Instances eines Clusters zu löschen, wird der Cluster nicht gelöscht. Sie müssen auch den Cluster löschen, aber dies ist nur möglich, wenn der Löschschutz deaktiviert ist.

Um festzustellen, ob der Löschschutz für den Cluster aktiviert ist, führen Sie den folgenden Befehl aus.

> **Tip**
>
> Um den Löschschutzstatus all Ihrer Amazon DocumentDB-Cluster anzuzeigen, lassen Sie den `--db-cluster-identifier` Parameter weg.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterIdentifier,DeletionProtection]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-cluster",
        "true"
    ]
]
```

Wenn für einen Cluster der Löschschutz aktiviert ist, bearbeiten Sie den Cluster, um den Löschschutz zu deaktivieren. Führen Sie den folgenden Befehl aus, um den Löschschutz des Clusters zu deaktivieren.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --no-deletion-protection \
    --apply-immediately
```

3. Löschen Sie schließlich den Cluster.

   Nachdem Sie den Löschschutz deaktiviert haben, können Sie den Cluster löschen. Um einen Cluster zu löschen, verwenden Sie die `delete-db-cluster`-Operation mit den folgenden Parametern.

   - **`--db-cluster-identifier`**—Erforderlich. Die ID des Clusters, den Sie löschen möchten.
   - **`--final-db-snapshot-identifier`**—Optional. Wenn Sie einen endgültigen Snapshot wünschen, müssen Sie diesen Parameter mit einem Namen für den endgültigen Snapshot angeben. Sie müssen entweder `--final-db-snapshot-identifier` oder `--skip-final-snapshot` angeben.

     Benennungseinschränkungen:

     - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
     - Muss mit einem Buchstaben beginnen.
     - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
     - Muss für alle Cluster in Amazon RDS, Amazon Neptune und Amazon DocumentDB pro und AWS-KontoRegion eindeutig sein.

- **`--skip-final-snapshot`**—Optional. Verwenden Sie diesen Parameter nur, wenn Sie vor dem Löschen Ihres Clusters keinen letzten Snapshot machen möchten. Standardmäßig wird ein letzter Snapshot erstellt. Sie müssen entweder `--final-db-snapshot-identifier` oder `--skip-final-snapshot` angeben.

Der folgende AWS CLI Code löscht den Cluster `sample-cluster` mit einem endgültigen Snapshot. Die Operation schlägt fehl, wenn Instances vorhanden sind, die mit dem Cluster verknüpft sind, oder wenn der Löschschutz aktiviert ist.

Example

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Für Windows:

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Example

Der folgende AWS CLI Code löscht den Cluster, `sample-cluster` ohne einen endgültigen Snapshot zu erstellen.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

Für Windows:

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
```

```
    --skip-final-snapshot
```

Die Ausgabe der `delete-db-cluster`-Operation ist der Cluster, den Sie löschen.

Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Status des Clusters finden Sie unter Den Status eines Clusters überwachen.

Skalieren von Amazon DocumentDB-Clustern

Mit Amazon DocumentDB können Sie den Speicher und die Rechenleistung in Ihren Clustern entsprechend Ihren Anforderungen skalieren. In diesem Abschnitt wird beschrieben, wie Sie Speicherskalierung, Instance-Skalierung und Leseskalierung verwenden können, um die Leistung und Skalierung für Ihre Amazon DocumentDB-Cluster und -Instances zu verwalten.

Themen

Speicherskalierung

Der Amazon DocumentDB-Speicher wird automatisch mit den Daten in Ihrem Cluster-Volume skaliert. Wenn Ihre Daten zunehmen, wächst Ihr Cluster-Volume-Speicher in Schritten von 10 GiB auf bis zu 128 TiB.

Skalierung von Instances

Sie können Ihren Amazon DocumentDB-Cluster nach Bedarf skalieren, indem Sie die Instance-Klasse für jede Instance im Cluster ändern. Amazon DocumentDB unterstützt mehrere Instance-Klassen, die für Amazon DocumentDB optimiert sind.

Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB-Instance.

Skalierung von Lesevorgängen

Sie können die Leseskalierung für Ihren Amazon DocumentDB-Cluster erreichen, indem Sie bis zu 15 Amazon DocumentDB-Replikate im Cluster erstellen. Jedes Amazon DocumentDB-Replikat gibt

dieselben Daten aus dem Cluster-Volume mit minimaler Replikatverzögerung zurück – in der Regel weniger als 100 Millisekunden, nachdem die primäre Instance ein Update geschrieben hat. Wenn Ihr Lesedatenverkehr zunimmt, können Sie zusätzliche Amazon DocumentDB-Replikate erstellen und direkt eine Verbindung zu ihnen herstellen, um die Leselast für Ihren Cluster zu verteilen. Amazon DocumentDB-Replikate müssen nicht derselben Instance-Klasse angehören wie die primäre Instance.

Weitere Informationen finden Sie unter Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster.

Um die Skalierung mit Amazon DocumentDB zu lesen, empfehlen wir Ihnen, eine Verbindung zu Ihrem Cluster als Replikatsatz herzustellen und Lesevorgänge mithilfe der integrierten Lesepräferenzfunktionen Ihres Treibers an Replikat-Instances zu verteilen. Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz.

Schreibskalierung

Sie können die Schreibkapazität auf Ihrem Amazon DocumentDB-Cluster skalieren, indem Sie die Größe der primären Instance Ihres Clusters erhöhen. In diesem Abschnitt werden zwei Methoden für die Skalierung der primären Instance Ihres Clusters abhängig von Ihren Anforderungen beschrieben. Die erste Option ermöglicht Ihnen die Minimierung von Auswirkungen auf Anwendungen, erfordert jedoch eine größere Zahl von Schritten. Die zweite Option ermöglicht Ihnen die Optimierung in Bezug auf die Einfachheit, da sie weniger Schritte erfordert. Die potenziellen Auswirkungen auf Anwendungen sind jedoch größer.

Abhängig von Ihrer Anwendung können Sie wählen, welcher Ansatz für Sie am besten geeignet ist. Weitere Informationen zu verfügbaren Instance-Größen und -Kosten finden Sie auf der Seite Amazon DocumentDB – Preise.

1. Optimieren für hohe Verfügbarkeit und Leistung – Wenn Sie eine Verbindung zu Ihrem Cluster im Replikat-Set-Modus herstellen (empfohlen), können Sie den folgenden Prozess verwenden, um die Auswirkungen auf Ihre Anwendung beim Skalieren Ihrer primären Instance zu minimieren. Diese Methode minimiert die Auswirkungen, da Ihr Cluster den von Ihnen festgelegten Wert für Hochverfügbarkeit weiter einhält oder überschreitet. Darüber hinaus werden Leseskalierungsziele dem Cluster als Instances hinzugefügt, anstatt an Ort und Stelle aktualisiert zu werden.

    a. Fügen Sie ein oder mehrere Replikate des größeren Instance-Typs zu Ihrem Cluster hinzu (siehe ???). Alle Replikate sollten einen Instance-Typ besitzen, der größer oder gleich

dem Typ der primären Instance ist. Dies vermeidet eine unbeabsichtigte Reduzierung der Schreibleistung aufgrund eines Failovers zu einem kleineren Instance-Typ. Für die meisten Kunden bedeutet dies, dass die Anzahl der Instances in ihrem Cluster vorübergehend verdoppelt wird und die kleineren Replikate nach Abschluss der Skalierung entfernt werden.

b. Legen Sie die Failover-Stufe für alle neuen Replikate auf die Priorität Null fest. Dies stellt sicher, dass ein Replikat des kleineren Instance-Typs die höchste Failover-Priorität besitzt. Weitere Informationen finden Sie unter ???.

c. Initiieren Sie einen manuellen Failover. Hierdurch wird eines der neuen Replikate zur primären Instance hochgestuft. Weitere Informationen finden Sie unter ???.

> **Note**
>
> Dies führt zu einer Ausfallzeit von ca. 30 Sekunden für Ihren Cluster. Bitte planen Sie entsprechend.

d. Entfernen Sie aus dem Cluster alle Replikate von Instance-Typen, die kleiner als der neue primäre Instance-Typ sind.

e. Setzen Sie die Failover-Stufe aller Instances auf dieselbe Priorität zurück. (In der Regel bedeutet dies, dass sie auf 1 zurückgesetzt werden.)

Angenommen, Sie führen einen Cluster aus, der zurzeit drei `r5.large`-Instances enthält (eine primäre Instance und zwei Replikate). Sie möchten diese Instances auf den Instance-Typ `r5.xlarge` skalieren. Hierzu fügen Sie dem Cluster zunächst drei `r5.xlarge`-Replikat-Instances hinzu und legen anschließend die Failover-Stufe der neuen `r5.xlarge`-Replikate auf null fest. Als Nächstes leiten Sie einen manuellen Failover ein. (Ihnen ist bekannt, dass dies zu einer Ausfallzeit von ca. 30 Sekunden für Ihre Anwendung führt.) Nach Abschluss des Failovers entfernen Sie alle drei `r5.large`-Instances aus dem Cluster. Der Cluster ist nun auf `r5.xlarge`-Instances skaliert.

Zur Kostenoptimierung werden Amazon DocumentDB-Instances in Schritten von einer Sekunde abgerechnet, wobei nach einer abrechenbaren Statusänderung wie dem Erstellen, Ändern oder Löschen einer Instance eine Mindestgebühr von zehn Minuten berechnet wird. Weitere Informationen finden Sie unter Kostenoptimierung in der Dokumentation zu bewährten Methoden.

2. Optimieren der Einfachheit – Dieser Ansatz optimiert die Einfachheit. Der Cluster wird nicht erweitert und komprimiert, aber er kann Ihre Lesekapazität vorübergehend reduzieren.

Es ist möglich, dass das Ändern der Instance-Klasse eines Replikats dazu führt, dass diese Instance für einen kurzen Zeitraum keine Anfragen bedient, von einigen Sekunden bis zu weniger als 30 Sekunden. Wenn Sie eine Verbindung zu Ihrem Cluster im Replikat-Set-Modus herstellen (empfohlen), würde dies Ihre Lesekapazität während des Skalierungsvorgangs um ein Replikat reduzieren (z. B. auf 66 % Kapazität in einem Cluster mit 3 Knoten oder 75 % Kapazität in einem Cluster mit 4 Knoten usw.).

a. Skalieren Sie eine der Replikat-Instances in Ihrem Cluster. Weitere Informationen finden Sie unter Verwalten von Instance-Klassen.

b. Warten Sie, bis die Instance verfügbar ist (siehe Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance).

> **Note**
>
> Dies führt zu einer Ausfallzeit von ca. 30 Sekunden für Ihren Cluster. Bitte planen Sie entsprechend.

c. Fahren Sie mit der Ausführung der Schritte 1 und 2 fort, bis alle Replikat-Instances nacheinander skaliert wurden.

d. Initiieren Sie ein manuelles Failover. Dadurch wird eines der Replikate zur primären Instance hochgestuft. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Failover.

> **Note**
>
> Dies führt zu einer Ausfallzeit von bis zu 30 Sekunden für Ihren Cluster, dauert aber oft weniger Zeit. Bitte planen Sie entsprechend.

e. Skalieren Sie die frühere primäre (jetzt ein Replikat) Instance.

Klonen eines Volumes für einen Amazon DocumentDB-Cluster

Durch die Verwendung des Klonens von Amazon DocumentDB können Sie einen neuen Cluster erstellen, der dasselbe Amazon DocumentDB-Cluster-Volume und dieselben Daten wie das Original enthält. Der Prozess ist so konzipiert, dass er schnell und kostengünstig ist. Der neue Cluster mit dem zugehörigen Datenvolume wird als clone (Klon) bezeichnet. Das Erstellen eines Klons ist

schneller und platzsparender als das physische Kopieren der Daten mit anderen Techniken, wie z. B. das Wiederherstellen eines Snapshots.

Amazon DocumentDB unterstützt das Erstellen eines von Amazon DocumentDB bereitgestellten Klons aus einem bereitgestellten Amazon DocumentDB-Cluster. Wenn Sie einen Klon mit einer anderen Bereitstellungskonfiguration als der Quelle erstellen, wird der Klon mit der neuesten Version der Amazon DocumentDB-Engine der Quelle erstellt.

Wenn Sie Klone aus Ihren Amazon DocumentDB-Clustern erstellen, werden die Klone in Ihrem AWS Konto erstellt – demselben Konto, dem der Amazon DocumentDB-Quellcluster gehört.

Themen

Übersicht über das Klonen von Amazon DocumentDB

Amazon DocumentDB verwendet ein copy-on-write Protokoll, um einen Klon zu erstellen. Dieser Mechanismus verwendet minimalen zusätzlichen Speicherplatz, um einen ersten Klon zu erstellen. Wenn der Klon zum ersten Mal erstellt wird, behält Amazon DocumentDB eine einzelne Kopie der Daten bei, die vom Quell-DB-Cluster und dem neuen (klonten) Amazon DocumentDB-Cluster verwendet werden. Zusätzlicher Speicher wird nur zugewiesen, wenn Änderungen an Daten (auf dem Amazon DocumentDB-Speichervolume) durch den Amazon DocumentDB-Quellcluster oder den Amazon DocumentDB-Cluster-Klon vorgenommen werden. Weitere Informationen zum copy-on-write Protokoll finden Sie unter Funktionsweise des Klonens von Amazon DocumentDB .

Das Klonen von Amazon DocumentDB ist besonders nützlich, um Testumgebungen mithilfe Ihrer Produktionsdaten schnell einzurichten, ohne dass Daten beschädigt werden. Sie können Klone für viele Arten von Anwendungen verwenden, z. B. für Folgende:

- Experimentieren Sie mit möglichen Änderungen (z. B. Schemaänderungen und Parametergruppenänderungen), um alle Auswirkungen zu bewerten.
- Führen Sie Workload-intensive Vorgänge aus, z. B. das Exportieren von Daten oder das Ausführen analytischer Abfragen auf dem Klon.

- Erstellen Sie eine Kopie Ihres Produktions-DB-Clusters zu Entwicklungs-, Test- oder anderen Zwecken.

Sie können mehr als einen Klon aus demselben Amazon DocumentDB-Cluster erstellen. Sie können auch mehrere Klone aus einem anderen Klon erstellen.

Nachdem Sie einen Amazon DocumentDB-Klon erstellt haben, können Sie die Amazon DocumentDB-Instances anders als den Amazon DocumentDB-Quellcluster konfigurieren. Beispielsweise benötigen Sie möglicherweise keinen Klon für Entwicklungszwecke, um dieselben Hochverfügbarkeitsanforderungen wie der Amazon DocumentDB-Quellproduktionscluster zu erfüllen. In diesem Fall können Sie den Klon mit einer einzelnen Amazon DocumentDB-Instance konfigurieren und nicht mit mehreren DB-Instances, die vom Amazon DocumentDB-Cluster verwendet werden.

Wenn Sie den Klon für Test-, Entwicklungs- oder andere Zwecke nicht mehr verwenden, können Sie ihn löschen.

Einschränkungen beim Klonen von Amazon DocumentDB

Amazon DocumentDB; das Klonen hat derzeit die folgenden Einschränkungen:

- Sie können so viele Klone erstellen, wie Sie möchten, bis zur maximalen Anzahl von DB-Clustern, die in der AWS-Region zulässig sind. Nachdem Sie jedoch 15 Klone erstellt haben, ist der nächste Klon eine vollständige Kopie. Der Klonvorgang funktioniert wie eine point-in-time Wiederherstellung.
- Sie können keinen Klon in einer anderen AWS Region als dem Amazon DocumentDB-Quellcluster erstellen.
- Sie können keinen Klon aus einem Amazon DocumentDB-Cluster ohne DB-Instances erstellen. Sie können nur Amazon DocumentDB-Cluster klonen, die mindestens eine DB-Instance haben.
- Sie können einen Klon in einer anderen Virtual Private Cloud (VPC) als der des Amazon DocumentDB-Clusters erstellen. In diesem Fall müssen die Subnetze der VPCs denselben Availability Zones zugeordnet sein.

Funktionsweise des Klonens von Amazon DocumentDB

Das Klonen von Amazon DocumentDB funktioniert auf der Speicherebene eines Amazon DocumentDB-Clusters. Es verwendet ein copy-on-write Protokoll, das sowohl in Bezug auf die zugrunde liegenden dauerhaften Medien, die das Amazon DocumentDB-Speichervolumen

unterstützen, schnell als auch platzsparend ist. Weitere Informationen zu Amazon DocumentDB-Cluster-Volumes finden Sie unter Verwalten von Amazon DocumentDB-Clustern.

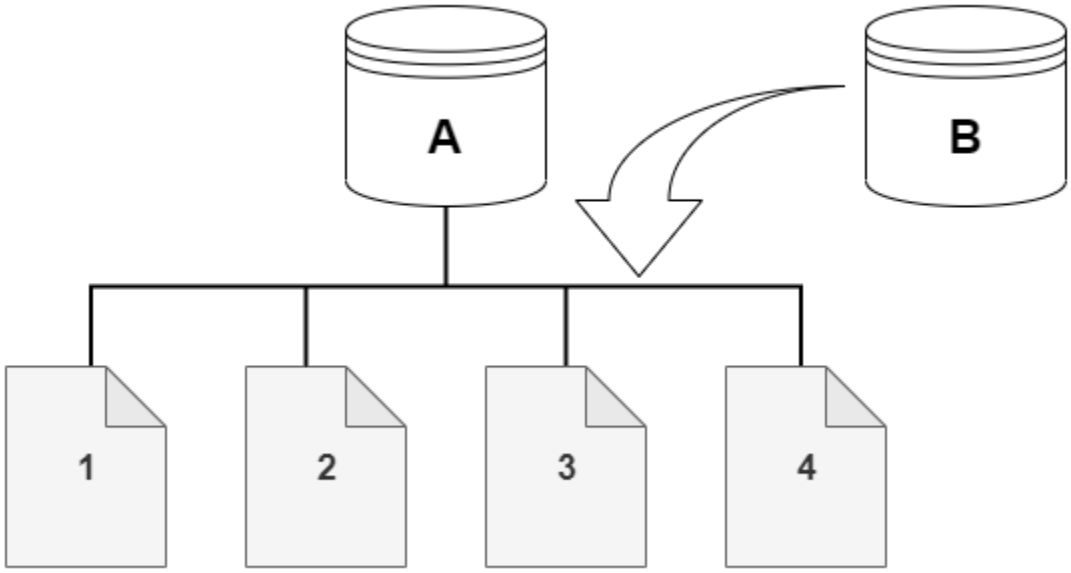
Themen

- Grundlegendes zum copy-on-write Protokoll
- Löschen eines Quell-Cluster-Volumes

Grundlegendes zum copy-on-write Protokoll

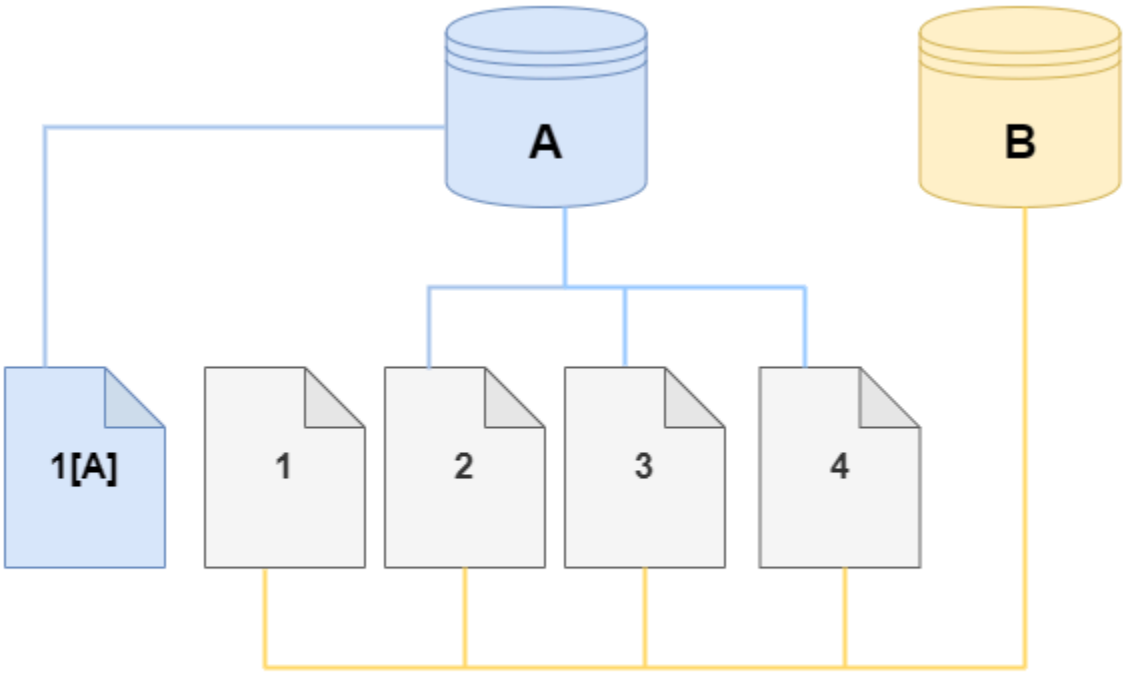
Ein Amazon DocumentDB-Cluster speichert Daten in Seiten im zugrunde liegenden Amazon DocumentDB-Speichervolume.

Im folgenden Diagramm finden Sie beispielsweise einen Amazon DocumentDB-Cluster (A) mit vier Datenseiten, 1, 2, 3 und 4. Angenommen, ein Klon, B, wird aus dem Amazon DocumentDB-Cluster erstellt. Wenn der Klon erstellt wird, werden keine Daten kopiert. Stattdessen verweist der Klon auf denselben Satz von Seiten wie der Amazon DocumentDB-Quell-Cluster.

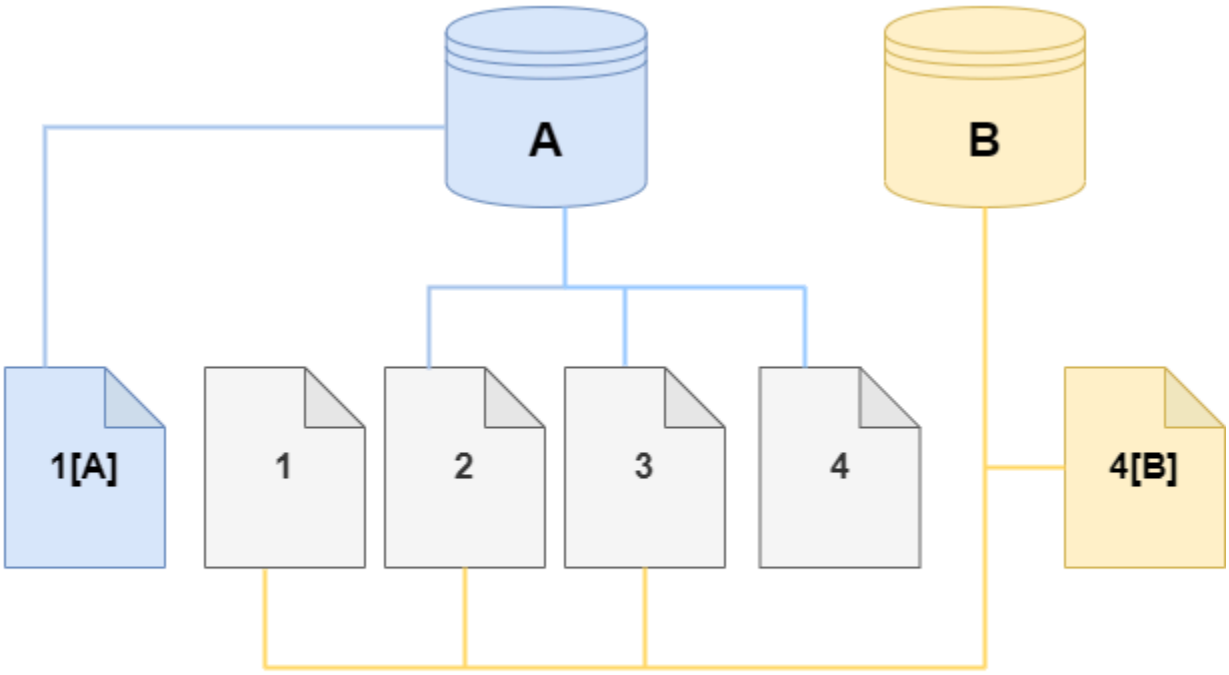


Wenn der Klon erstellt wird, ist normalerweise kein zusätzlicher Speicher erforderlich. Das copy-on-write Protokoll verwendet dasselbe Segment auf den physischen Speichermedien wie das Quellsegment. Zusätzlicher Speicher ist nur erforderlich, wenn die Kapazität des Quellsegments für das gesamte Klonsegment nicht ausreicht. Wenn dies der Fall ist, wird das Quellsegment auf ein anderes physisches Gerät kopiert.

In den folgenden Diagrammen finden Sie ein Beispiel für das copy-on-write Protokoll in Aktion mit demselben Cluster A und seinem Klon, B, wie oben gezeigt. Nehmen wir an, Sie nehmen eine Änderung an Ihrem Amazon DocumentDB-Cluster (A) vor, was zu einer Änderung der auf Seite 1 gespeicherten Daten führt. Anstatt auf die ursprüngliche Seite 1 zu schreiben, erstellt Amazon DocumentDB eine neue Seite 1 [A]. Das Amazon DocumentDB-Cluster-Volume für Cluster (A) verweist jetzt auf Seite 1[A], 2, 3 und 4, während der Klon (B) weiterhin auf die ursprünglichen Seiten verweist.



Auf dem Klon wird eine Änderung an Seite 4 auf dem Speichervolume vorgenommen. Anstatt auf die ursprüngliche Seite 4 zu schreiben, erstellt Amazon DocumentDB eine neue Seite, 4 [B]. Der Klon verweist nun auf die Seiten 1, 2, 3 und auf Seite 4[B], während der Cluster (A) weiterhin auf 1[A], 2, 3 und 4 verweist.



Wenn im Laufe der Zeit mehr Änderungen sowohl im Quell-Volume des Amazon DocumentDB-Clusters als auch im Klon auftreten, wird mehr Speicher benötigt, um die Änderungen zu erfassen und zu speichern.

Löschen eines Quell-Cluster-Volumes

Wenn Sie ein Quell-Cluster-Volume löschen, dem ein oder mehrere Klone zugeordnet sind, sind die Klone nicht betroffen. Die Klone verweisen weiter auf die Seiten, die zuvor im Besitz des Quell-Cluster-Volumes waren.

Erstellen eines Amazon DocumentDB-Klons

Sie können einen Klon im selben AWS Konto wie der Amazon DocumentDB-Quellcluster erstellen. Dazu können Sie die AWS Management Console oder die AWS CLI und die folgenden Verfahren verwenden.

Mit dem Klonen von Amazon DocumentDB können Sie einen bereitgestellten Amazon DocumentDB-Cluster-Klon aus einem bereitgestellten Amazon DocumentDB-Cluster erstellen.

Using the AWS Management Console

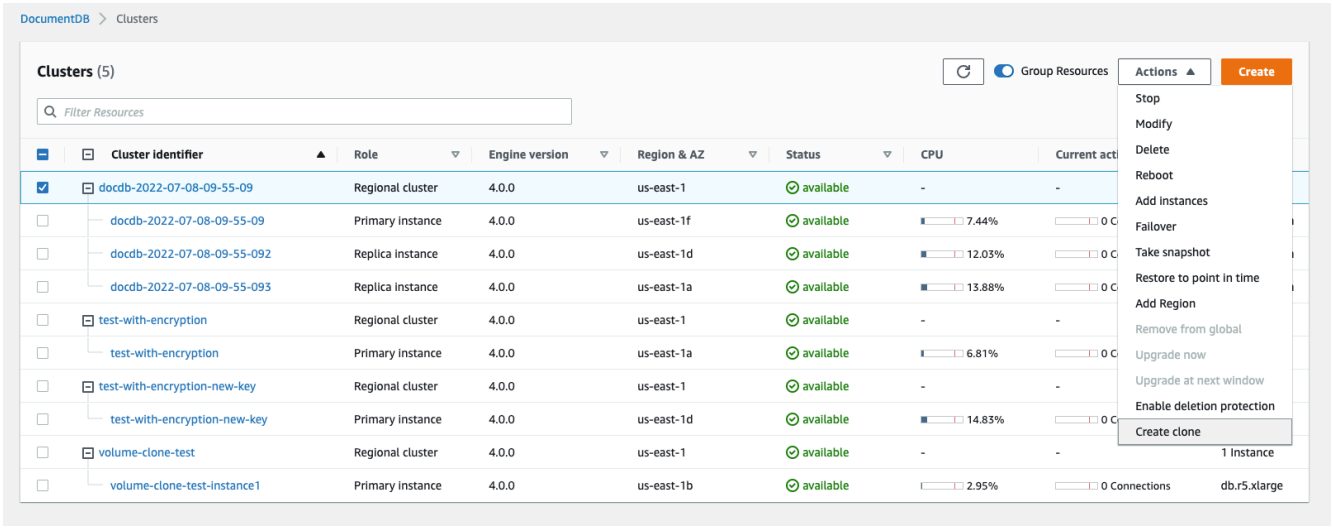
Im folgenden Verfahren wird beschrieben, wie Sie einen Amazon DocumentDB-Cluster mithilfe der klonen AWS Management Console.

Erstellen eines Klons mithilfe der AWS Management Console Ergebnisse in einem Amazon DocumentDB-Cluster mit einer Amazon DocumentDB-Instance.

Diese Anweisungen gelten für DB-Cluster, die demselben - AWS Konto gehören, das den Klon erstellt. Der DB-Cluster muss demselben - AWS Konto gehören, da das kontoübergreifende Klonen in Amazon DocumentDB nicht unterstützt wird.

So erstellen Sie einen Klon eines DB-Clusters, der Ihrem AWS Konto gehört, mithilfe der AWS Management Console

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie Ihren Amazon DocumentDB-Cluster aus der Liste aus und wählen Sie für Aktionen die Option Klon erstellen aus.

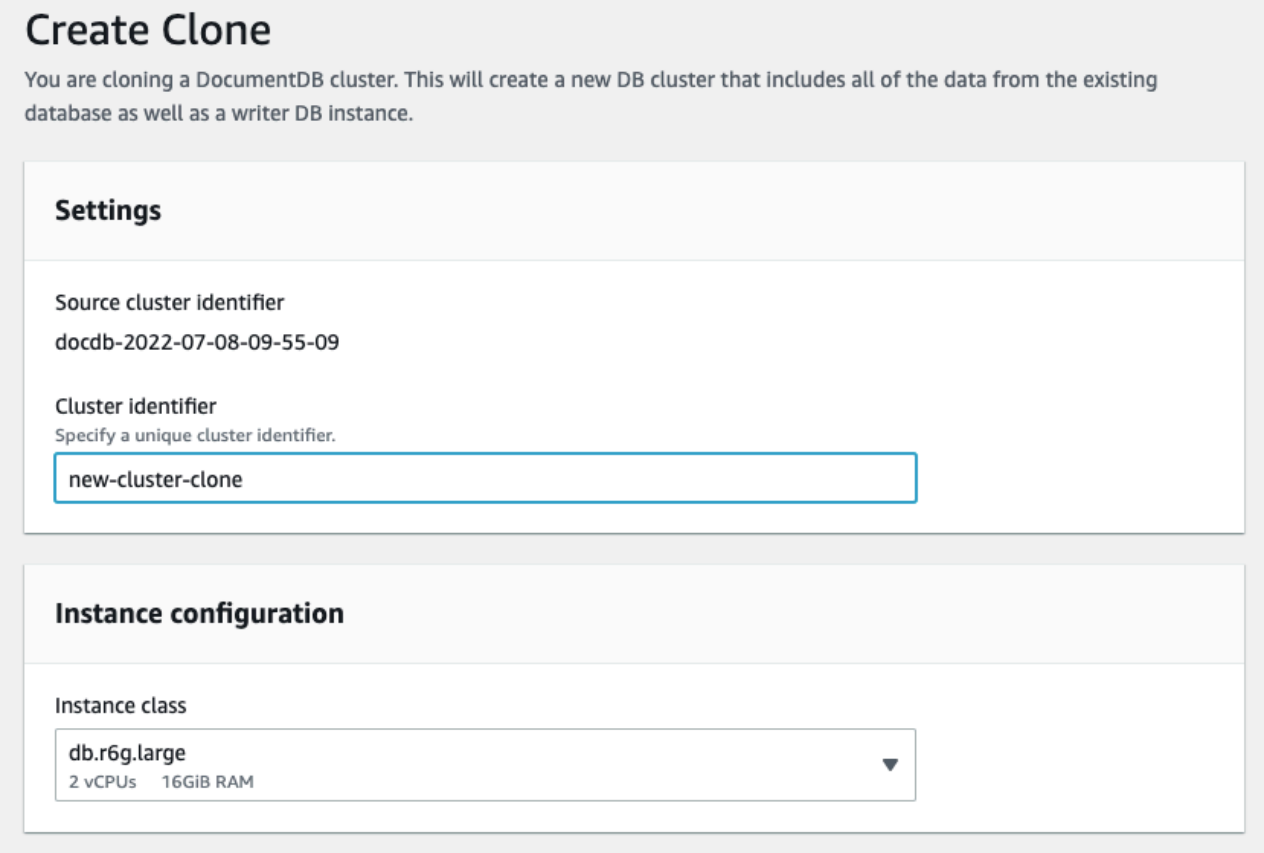


Die Seite Klon erstellen wird geöffnet, auf der Sie eine Cluster-ID und eine Instance-Klasse sowie andere Optionen für den Amazon DocumentDB-Cluster-Klon konfigurieren können.

4. Gehen Sie im Abschnitt Settings (Einstellungen) wie folgt vor:

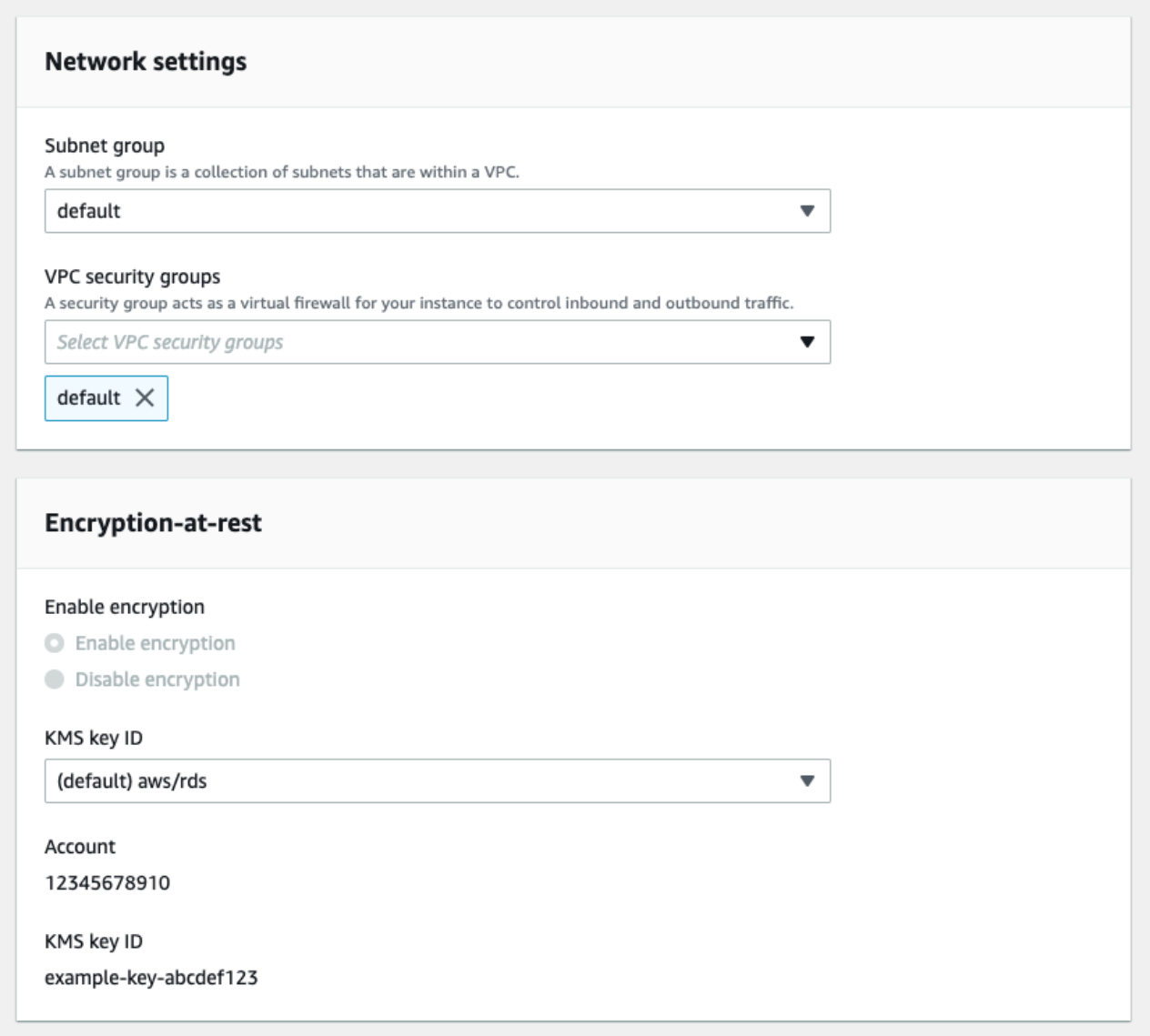
   a. Geben Sie für Cluster-ID den Namen ein, den Sie Ihrem geklonten Amazon DocumentDB-Cluster geben möchten.

b. Wählen Sie für Instance-Konfiguration eine geeignete Instance-Klasse für Ihren geklonten Amazon DocumentDB-Cluster aus.

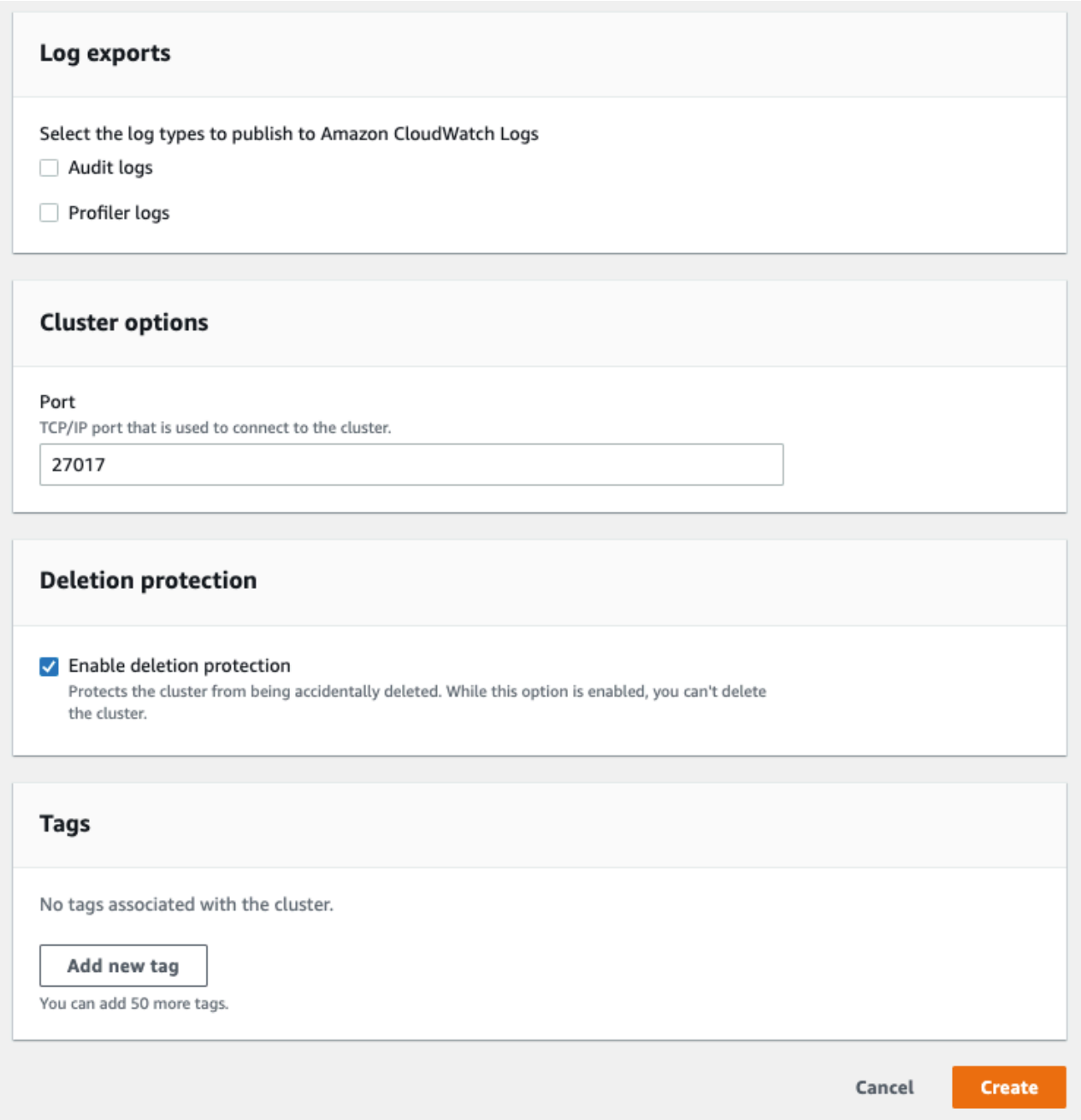


c. Wählen Sie für Netzwerkeinstellungen eine Subnetzgruppe für Ihren Anwendungsfall und die zugehörigen VPC-Sicherheitsgruppen aus.

d. Wenn für Encryption-at-rest für den Quell-Cluster (den Cluster, der geklont wird) die Verschlüsselung aktiviert ist, muss für den geklonten Cluster auch die Verschlüsselung aktiviert sein. Wenn dieses Szenario zutrifft, sind die Optionen Verschlüsselung aktivieren ausgegraut (deaktiviert), aber die Option Verschlüsselung aktivieren ausgewählt. Umgekehrt, wenn für den Quell-Cluster keine Verschlüsselung aktiviert ist, sind die Optionen Verschlüsselung aktivieren verfügbar und Sie können die Verschlüsselung aktivieren oder deaktivieren.



e. Schließen Sie die neue Cluster-Klonkonfiguration ab, indem Sie den Typ der zu exportierenden Protokolle auswählen (optional), einen bestimmten Port eingeben, der für die Verbindung mit dem Cluster verwendet wird, und den Schutz vor versehentlichem Löschen des Clusters aktivieren (standardmäßig aktiviert).



f. Schließen Sie die Eingabe aller Einstellungen für Ihren Amazon DocumentDB-Cluster-Klon ab. Weitere Informationen zu Amazon DocumentDB-Cluster- und Instance-Einstellungen finden Sie unter Verwalten von Amazon DocumentDB-Clustern.

5. Wählen Sie Klon erstellen, um den Amazon DocumentDB-Klon des von Ihnen ausgewählten Amazon DocumentDB-Clusters zu starten.

Wenn der Klon erstellt wird, wird er mit Ihren anderen Amazon DocumentDB-Clustern im Abschnitt Datenbanken der Konsole aufgeführt und zeigt seinen aktuellen Status an. Ihr Klon ist einsatzbereit, wenn sein Status Verfügbar ist.

Using the AWS CLI

Die Verwendung der AWS CLI zum Klonen Ihres Amazon DocumentDB-Clusters umfasst einige Schritte.

Der von Ihnen verwendete `restore-db-cluster-to-point-in-time` AWS CLI Befehl führt zu einem leeren Amazon DocumentDB-Cluster mit 0 Amazon DocumentDB-Instances. Das heißt, der Befehl stellt nur den Amazon DocumentDB-Cluster wieder her, nicht die DB-Instances für diesen Cluster. Sie tun dies separat, nachdem der Klon verfügbar ist. Die zwei Schritte im Prozess sind wie folgt:

1. Erstellen Sie den Klon mit dem CLI-Befehl restore-db-cluster-to-point-in-time. Die Parameter, die Sie mit diesem Befehl verwenden, steuern den Kapazitätstyp und andere Details des leeren Amazon DocumentDB-Clusters (Klons), der erstellt wird.
2. Erstellen Sie die Amazon DocumentDB-Instance für den Klon, indem Sie den create-db-instance CLI-Befehl verwenden, um die Amazon DocumentDB-Instance im wiederhergestellten Amazon DocumentDB-Cluster neu zu erstellen.

Bei den folgenden Befehlen AWS CLI wird davon ausgegangen, dass mit Ihrer AWS Region als Standard eingerichtet ist. Dieser Ansatz erspart Ihnen die Übergabe des `--region`-Namens in jedem der Befehle. Weitere Informationen finden Sie unter Konfigurieren der AWS CLI. Sie können die `--region` auch in jedem der folgenden CLI-Befehle angeben.

Erstellen des Klons

Die spezifischen Parameter, die Sie an den `restore-db-cluster-to-point-in-time`-CLI-Befehl übergeben, variieren. Was Sie übergeben, hängt vom Typ des Klons ab, den Sie erstellen möchten.

Gehen Sie wie folgt vor, um einen bereitgestellten Amazon DocumentDB-Klon aus einem bereitgestellten Amazon DocumentDB-Cluster zu erstellen.

So erstellen Sie einen Klon im gleichen Engine-Modus wie der Amazon DocumentDB-Quell-Cluster

- Verwenden Sie den `restore-db-cluster-to-point-in-time`-CLI-Befehl und geben Sie Werte für die folgenden Parameter an:

  - `--db-cluster-identifier` – Wählen Sie einen aussagekräftigen Namen für Ihren Klon. Sie benennen den Klon, wenn Sie den restore-db-cluster-to-point-in-timeCLI-Befehl verwenden.
  - `--restore-type` – Verwenden Sie `copy-on-write`, um einen Klon des Quell-DB-Clusters zu erstellen. Ohne diesen Parameter stellt das den Amazon DocumentDB-Cluster `restore-db-cluster-to-point-in-time` wieder her, anstatt einen Klon zu erstellen. Der Standardwert für `restore-type` ist `full-copy`.
  - `--source-db-cluster-identifier` – Verwenden Sie den Namen des Amazon DocumentDB-Quell-Clusters, den Sie klonen möchten.
  - `--use-latest-restorable-time` – Dieser Wert verweist auf die neuesten wiederherstellbaren Volume-Daten für den Klon. Dieser Parameter ist für erforderlich, Sie können `restore-type copy-on-write`jedoch nicht `restore-to-time parameter` mit verwenden.

Im folgenden Beispiel wird ein Klon namens `my-clone` aus einem Cluster namens `my-source-cluster` erstellt.

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier my-source-cluster \
    --db-cluster-identifier my-clone \
    --restore-type copy-on-write \
    --use-latest-restorable-time
```

Für Windows:

```
aws docdb restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier my-source-cluster ^
    --db-cluster-identifier my-clone ^
    --restore-type copy-on-write ^
    --use-latest-restorable-time
```

Der Befehl gibt das JSON-Objekt zurück, das Details des Klons enthält. Stellen Sie sicher, dass Ihr geklonter DB-Cluster verfügbar ist, bevor Sie versuchen, die DB-Instance für Ihren Klon zu erstellen. Weitere Informationen finden Sie unten unter Überprüfen des Status und Abrufen von Klondetails:

Überprüfen des Status und Abrufen von Klondetails

Mit dem folgenden Befehl können Sie den Status Ihres neu erstellten leeren DB-Clusters überprüfen.

```
$ aws docdb describe-db-clusters --db-cluster-identifier my-clone --query '*[].
[Status]' --output text
```

Sie können auch den Status und die anderen Werte abrufen, die Sie zum Erstellen der DB-Instance für Ihren Klon benötigen, indem Sie die folgende AWS CLI Abfrage verwenden:

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone \
  --query '*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}'
```

Für Windows:

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone ^
  --query "*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}"
```

Diese Abfrage gibt eine Ausgabe ähnlich der folgenden zurück.

```
[
  {
        "Status": "available",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
    }
]
```

Erstellen der Amazon DocumentDB-Instance für Ihren Klon

Verwenden Sie den create-db-instance CLI-Befehl , um die DB-Instance für Ihren Klon zu erstellen.

Der `--db-instance-class` Parameter wird nur für bereitgestellte Amazon DocumentDB-Cluster verwendet.

Für Linux, macOS oder Unix:

```
aws docdb create-db-instance \
    --db-instance-identifier my-new-db \
    --db-cluster-identifier my-clone \
    --db-instance-class  db.r5.4xlarge \
    --engine docdb
```

Für Windows:

```
aws docdb create-db-instance ^
    --db-instance-identifier my-new-db ^
    --db-cluster-identifier my-clone ^
    --db-instance-class  db.r5.4xlarge ^
    --engine docdb
```

Parameter, die zum Klonen verwendet werden sollen

Die folgende Tabelle fasst die verschiedenen Parameter zusammen, die mit `restore-db-cluster-to-point-in-time` zum Klonen von Amazon DocumentDB-Clustern verwendet werden.

Parameter	Beschreibung
--source-db-cluster-identifier	Verwenden Sie den Namen des Amazon DocumentDB-Quell-Clusters, den Sie klonen möchten.
--db-cluster-identifier	Wählen Sie einen aussagekräftigen Namen für Ihren Klon. Sie benennen Ihren Klon mit dem `restore-db-cluster-to-point-in-time` -Befehl. Dann übergeben Sie diesen Namen an den `create-db-instance` -Befehl.
--restore-type	Geben Sie `copy-on-write` als an`--restore-type` , um einen Klon des Quell-DB-Clusters zu erstellen, anstatt den Amazon DocumentDB-Quell-Cluster wiederherzustellen.

Parameter	Beschreibung
--use-latest-restorable-time	ieser Wert verweist auf die neuesten wiederherstellbaren Volume-Daten für den Klon.

Grundlegendes zur Fehlertoleranz des Amazon DocumentDB-Clusters

Amazon DocumentDB-Cluster sind von vornherein fehlertolerant. Das Volume jedes Clusters erstreckt sich über mehrere Availability Zones in einem einzigen AWS-Regionund jede Availability Zone enthält eine Kopie der Volume-Daten des Clusters. Diese Funktionalität bedeutet, dass Ihr Cluster einen Ausfall der Availability Zone ohne Datenverlust und nur eine kurze Unterbrechung des Services tolerieren kann.

Wenn die primäre Instance in einem Cluster ausfällt, führt Amazon DocumentDB automatisch ein Failover auf eine von zwei Arten auf eine neue primäre Instance durch:

- Indem Sie ein vorhandenes Amazon DocumentDB-Replikat auf die neue primäre Instance hochstufen, die basierend auf der Einstellung für das Hochstufungskontingent jedes Replikats ausgewählt wurde, und dann einen Ersatz für die frühere primäre Instance erstellen. Ein Failover auf die Replikat-Instance dauert in der Regel weniger als 30 Sekunden. Während dieses Zeitraums kann es zu einer kurzen Unterbrechung der Lese- und Schreibvorgänge kommen. Um die Verfügbarkeit Ihres Clusters zu erhöhen, empfehlen wir Ihnen, mindestens ein Amazon DocumentDB-Replikat in zwei oder mehr verschiedenen Availability Zones zu erstellen.
- Über das Anlegen einer neuen primären Instance. Dies geschieht nur, wenn Sie keine Replikat-Instance in Ihrem Cluster haben und der Abschluss einige Minuten dauern kann.

Wenn der Cluster über ein oder mehrere Amazon DocumentDB-Replikate verfügt, wird ein Amazon DocumentDB-Replikat während eines Ausfallereignisses zur primären Instance hochgestuft. Ein Fehlerereignis hat eine kurze Unterbrechung zufolge, während die Lese- und Schreibvorgänge mit einer Ausnahme fehlschlagen. Jedoch wird der Service im Normalfall in weniger als 120 Sekunden und oft sogar schon nach 60 Sekunden wiederhergestellt. Um die Verfügbarkeit Ihres Clusters zu erhöhen, empfehlen wir Ihnen, mindestens ein Amazon DocumentDB-Replikat in zwei oder mehr verschiedenen Availability Zones zu erstellen.

Sie können die Reihenfolge anpassen, in der Ihre Amazon DocumentDB-Replikate nach einem Ausfall zur primären Instance hochgestuft werden, indem Sie jedem Replikat eine Priorität zuweisen. Prioritäten liegen im Bereich zwischen 0 als höchste Priorität und 15 als niedrigste Priorität. Wenn

die primäre Instance ausfällt, wird das Amazon DocumentDB-Replikat mit der höchsten Priorität zur neuen primären Instance hochgestuft. Sie können die Priorität eines Amazon DocumentDB-Replikats jederzeit ändern. Das Ändern der Priorität löst kein Failover aus. Sie können die `modify-db-instance`-Operation mit dem `--promotion-tier`-Parameter verwenden. Weitere Informationen zum Anpassen der Failover-Priorität einer Instance finden Sie unter Amazon DocumentDB DocumentDB-Failover.

Mehrere Amazon DocumentDB-Replikate können dieselbe Priorität haben, was zu Hochstufungsstufen führt. Wenn zwei oder mehr Amazon DocumentDB-Replikate dieselbe Priorität haben, wird das größte Replikat zum primären Replikat hochgestuft. Wenn zwei oder mehrere Amazon DocumentDB-Replikate dieselbe Priorität und Größe haben, wird ein beliebiges Replikat in derselben Hochstufungsstufe hochgestuft.

Wenn der Cluster keine Amazon DocumentDB-Replikate enthält, wird die primäre Instance während eines Fehlerereignisses neu erstellt. Ein Fehlerereignis hat eine Unterbrechung zufolge, während die Lese- und Schreibvorgänge mit einer Ausnahme fehlschlagen. Der Service wird wiederhergestellt, wenn die primäre Instance erstellt wird. Dies dauert im Normalfall weniger als 10 Minuten. Das Hochstufen eines Amazon DocumentDB-Replikats zur primären Instance ist viel schneller als das Erstellen einer neuen primären Instance.

Verwalten von Amazon DocumentDB-Instances

Die folgenden Themen enthalten Informationen, die Sie bei der Verwaltung Ihrer Amazon DocumentDB-Instances unterstützen. Sie umfassen Details über Instance-Klassen und Statusarten und das Erstellen, Löschen und Ändern einer Instance.

Themen

- Verwalten von Instance-Klassen
- Bestimmen des Status einer Instance
- Amazon DocumentDB-Instance-Lebenszyklus

Verwalten von Instance-Klassen

Die Instance-Klasse bestimmt die Rechen- und Speicherkapazität einer Amazon DocumentDB-Instance (mit MongoDB-Kompatibilität). Welche Instance-Klasse Sie benötigen, hängt von der benötigten Rechenleistung und dem Speicherbedarf ab.

Amazon DocumentDB unterstützt die Familien R4, R5, R6G, T3 und T4G der Instance-Klassen. Diese Klassen sind Instance-Klassen der aktuellen Generation, die für speicherintensive Anwendungen optimiert sind. Die Spezifikationen dieser Klassen finden Sie unter Instance-Klassen-Spezifikationen.

Themen

Bestimmen einer Instance-Klasse

Zum Bestimmen der Klasse einer Instance können Sie dieAWS Management Console verwenden oder die Operation`describe-db-instances` AWS CLI.

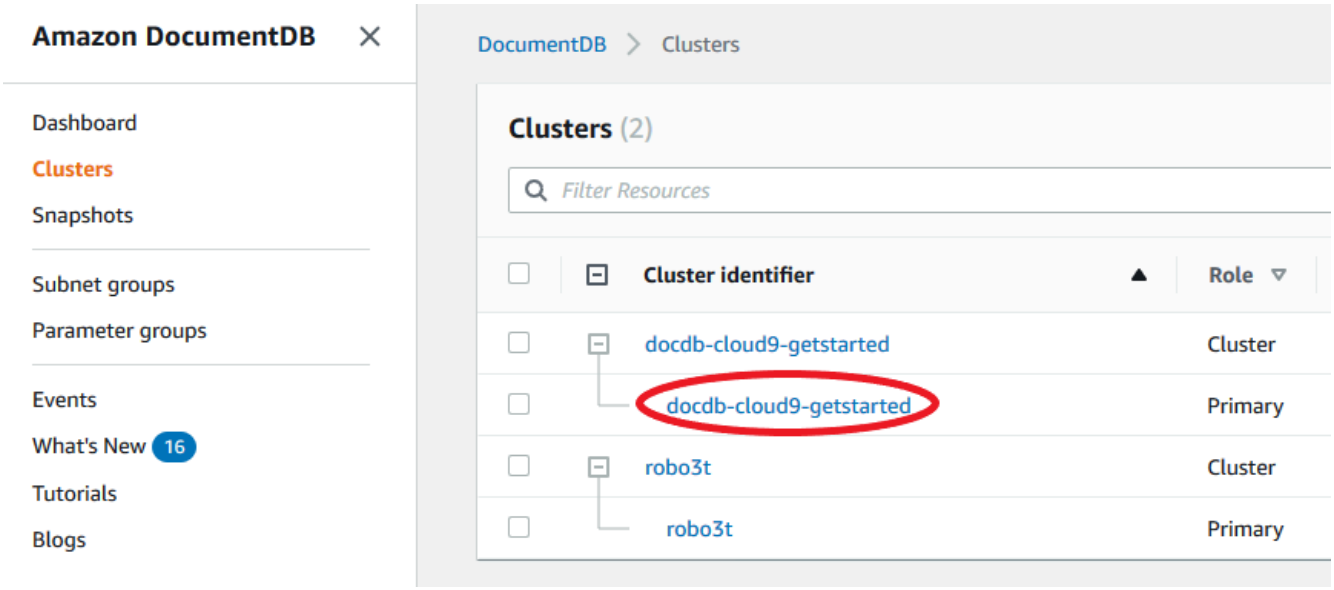
Using the AWS Management Console

Führen Sie die folgenden Schritte in der Konsole aus, um die Instance-Klasse für die Instances Ihres Clusters zu ermitteln.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus, um die gewünschte Instance zu finden.

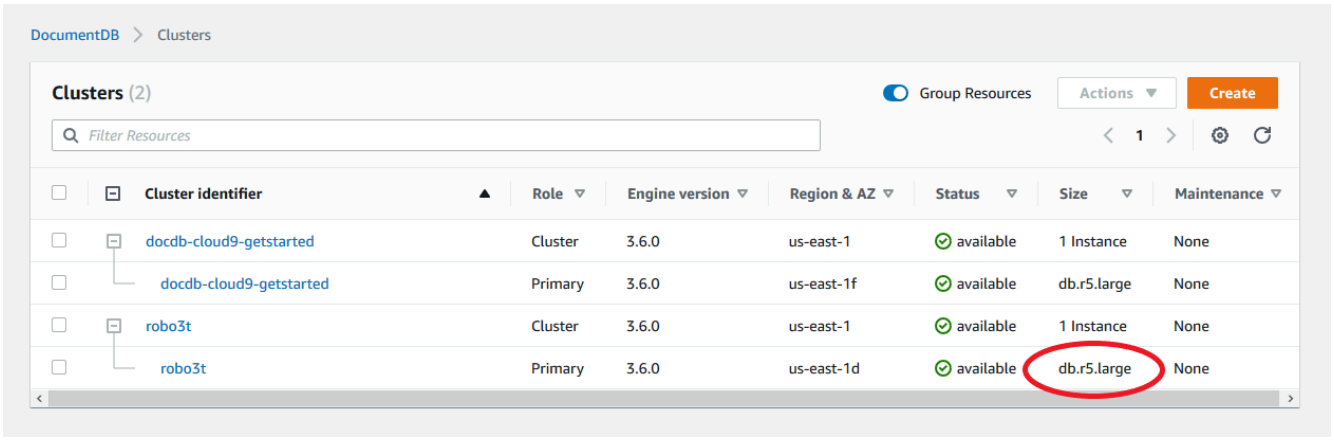
   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Navigationsfeld Cluster sehen Sie die Spalte Cluster-ID . Ihre Instances sind unter Cluster aufgeführt, ähnlich wie im folgenden Screenshot.



4. Erweitern Sie in der Liste der Instances den Cluster, um die Instances zu finden, an denen Sie interessiert sind. Suchen Sie die gewünschte Instance. Betrachten Sie dann die Spalte Größe der Zeile der Instance, um ihre Instance-Klasse anzuzeigen.

   In der folgenden Abbildung ist die Instance-Klasse für die Instance `robo3t` `db.r5.4xlarge`.



Using the AWS CLI

Um die Klasse einer Instance über die AWS CLI zu ermitteln, verwenden Sie die Operation `describe-db-instances` mit den folgenden Parametern.

- **`--db-instance-identifier`** – Optional. Gibt die Instance an, für die Sie die Instance-Klasse suchen möchten. Wenn dieser Parameter ausgelassen wird, gibt `describe-db-instances` eine Beschreibung für bis zu 100 Ihrer Instances zurück.

- **`--query`** – Optional. Gibt die Mitglieder der Instance an, die in die Ergebnisse einbezogen werden sollen. Wenn dieser Parameter ausgelassen wird, werden alle Instance-Mitglieder zurückgegeben.

Example

Im folgenden Beispiel werden der Instance-Name und die Klasse für die Instance ermitteltsample-instance-1.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --db-instance-identifier sample-instance-1
```

Für Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --db-instance-identifier sample-instance-1
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ]
```

Example

Im folgenden Beispiel werden der Instance-Name und die Klasse für bis zu 100 Amazon DocumentDB-Instances ermittelt.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --filter Name=engine,Values=docdb
```

Für Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --filter Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ],
    [
        "sample-instance-2",
        "db.r5.large"
    ],
    [
        "sample-instance-3",
        "db.r5.4xlarge"
    ],
    [
        "sample-instance-4",
        "db.r5.4xlarge"
    ]
]
```

Weitere Informationen finden Sie unter Beschreiben von Amazon DocumentDB-Instances.

Ändern einer Instance-Klasse

Sie können die Instance-Klasse Ihrer Instance mit der AWS Management Console oder AWS CLI ändern. Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB-Instance.

Unterstützte Instance-Klassen nach Region

Amazon DocumentDB unterstützt die folgenden Instance-Klassen:

- R6G– Die neueste Generation von arbeitsspeicheroptimierten Instances, die mit ARM-basierten AWS Graviton2-Prozessoren betrieben werden und eine bis zu 30 % bessere Leistung als R5-Instances zu 5 % günstigeren Kosten bieten.

- R5– Speicheroptimierte Instances, die zu denselben Instance-Kosten eine bis zu 100 % bessere Leistung als R4-Instances bieten.
- R4– Frühere Generierung von speicheroptimierten Instances.
- T4G– Kostengünstiger Burstable Allzweck-Instance-Typ der neuesten Generation mit ARM-basierten AWS Graviton2-Prozessoren, der eine CPU-Basisleistung bietet, eine bis zu 35 % bessere Preisleistung als T3-Instances bietet und ideal für die Ausführung von Anwendungen mit moderater CPU-Auslastung ist, bei denen es zu vorübergehenden Nutzungsspitzen kommt.
- T3– Kostengünstiger Allzweck-Instance-Typ, der eine CPU-Basisleistung mit der Möglichkeit bietet, die CPU-Auslastung jederzeit so lange wie nötig zu steigern.

Detaillierte Angaben zu den Instance-Klassen finden Sie unter Instance-Klassen-Spezifikationen.

Eine bestimmte Instance-Klasse kann in einer bestimmten Region unterstützt werden oder nicht. Die folgende Tabelle gibt an, welche Instance-Klassen von Amazon DocumentDB in jeder Region unterstützt werden.

Unterstützte Instance-Klassen nach Region

Region	R6G	R5	R4	T4G	T3
USA Ost (Ohio)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA Ost (Nord-Virginia)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA West (Oregon)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Südamerika (São Paulo)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hongkong)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hyderabad)		Unterstützt			Unterstützt

Region	R6G	R5	R4	T4G	T3
Asien-Pazifik (Mumbai)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Seoul)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Sydney)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Singapur)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Tokio)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Kanada (Zentral)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Frankfurt)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Irland)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Europa (London)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Milan)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Paris)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Region China (Peking)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
China (Ningxia)	Unterstützt	Unterstützt		Unterstützt	Unterstützt

Region	R6G	R5	R4	T4G	T3
AWS GovCloud (USA-West)	Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (USA-Ost)	Unterstützt	Unterstützt		Unterstützt	Unterstützt

Instance-Klassen-Spezifikationen

Die folgende Tabelle enthält Details zu den Amazon DocumentDB-Instance-Klassen. Erläuterungen zu den einzelnen Tabellenspalten finden Sie unterhalb der Tabelle.

Unterstützte Amazon DocumentDB-Instance-Klassen

Instance-Klasse	vCPU[1]	Speicher (GiB)[2]	Max. temporärer Speicher (GiB)[3]	Maximale Bandbreite (Mbit/s)[4]	Netzwerkperformance[5]	Unterstützung von Engines[6]
R6G – Speicheroptimierte Instance-Klasse der aktuellen Generation basierend auf Graviton2						
`db.r6g.large`	2	16	32	Bis zu 4.750.	Bis zu 10 Gbit/s	4.0.0 und 5.0.0
`db.r6g.xlarge`	4	32	63	Bis zu 4.750.	Bis zu 10 Gbit/s	4.0.0 und 5.0.0
`db.r6g.2xlarge`	8	64	126	Bis zu 4.750.	Bis zu 10 Gbit/s	4.0.0 und 5.0.0
`db.r6g.4xlarge`	16	128	252	4.750	Bis zu 10 Gbit/s	4.0.0 und 5.0.0
`db.r6g.8xlarge`	32	256	504	9 000	12 Gbit/s	4.0.0 und 5.0.0

Instance-Klasse	vCPU[1]	Speicher (GiB)[2]	Max. temporärer Speicher (GiB)[3]	Maximale Bandbreite (Mbit/s)[4]	Netzwerkperformance[5]	Unterstützung von Engines[6]
`db.r6g.12xlarge`	48	384	756	13.500	20 Gbit/s	4.0.0 und 5.0.0
`db.r6g.16xlarge`	64	512	1008	19.000	25 Gbit/s	4.0.0 und 5.0.0
R5 – Speicheroptimierte Instance-Klasse der vorherigen Generation						
`db.r5.large`	2	16	31	Bis zu 3.500	Bis zu 10 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.xlarge`	4	32	62	Bis zu 3.500	Bis zu 10 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.2xlarge`	8	64	124	Bis zu 3.500	Bis zu 10 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.4xlarge`	16	128	249	3.500	Bis zu 10 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.8xlarge`	32	256	504	6.800	10 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.12xlarge`	48	384	748	7.000	10 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.16xlarge`	64	512	1008	13.600	20 Gbit/s	3.6.0, 4.0.0 und 5.0.0
`db.r5.24xlarge`	96	768	1500	14.000	25 Gbit/s	3.6.0, 4.0.0 und 5.0.0
R4 – Speicheroptimierte Instance-Klasse der vorherigen Generation						

Instance-Klasse	vCPU[1]	Speicher (GiB)[2]	Max. temporärer Speicher (GiB)[3]	Maximale Bandbreite (Mbit/s)[4]	Netzwerkperformance[5]	Unterstützung von Engines[6]
db.r4.large	2	15,25	30	437	Bis zu 10 Gbit/s	Nur 3.6.0
db.r4.xlarge	4	30,5	60	875	Bis zu 10 Gbit/s	Nur 3.6.0
db.r4.2xlarge	8	61	120	875	Bis zu 10 Gbit/s	Nur 3.6.0
db.r4.4xlarge	16	122	240	875	Bis zu 10 Gbit/s	Nur 3.6.0
db.r4.8xlarge	32	244	480	875	10 Gbit/s	Nur 3.6.0
db.r4.16xlarge	64	488	960	14.000	25 Gbit/s	Nur 3.6.0
T4G – Instance-Klassen mit Spitzenlastleistung der neuesten Generation basierend auf Graviton2						
db.t4g.medium	2	4	8.13	Bis zu 2.085	Bis zu 5 GBit	4.0.0 und 5.0.0
T3 – Instance-Klassen mit Spitzenlastleistung der vorherigen Generation						
db.t3.medium	2	4	7,5	Bis zu 1.536	Bis zu 5 GBit	3.6.0, 4.0.0 und 5.0.0

Instance-Klasse	vCPU[1]	Speicher (GiB)[2]	Max. temporärer Speicher (GiB)[3]	Maximale Bandbreite (Mbit/s)[4]	Netzwerkperformance[5]	Unterstützung von Engines[6]

1. vCPU – Die Anzahl der virtuellen zentralen Verarbeitungseinheiten (CPUs). Eine virtuelle CPU ist eine Kapazitätseinheit, mit der Sie Instance-Klassen vergleichen können. Anstatt einen bestimmten Prozessor für mehrere Monate oder Jahre zu erwerben oder zu leasen, wird jetzt Kapazität stundenweise gemietet. Unser Ziel besteht darin, unabhängig von der tatsächlich zu Grunde liegenden Hardware eine gleichbleibende Menge an CPU-Kapazität zu bieten.
2. Speicher (GiB) – Der RAM in Gigabyte, der der Instance zugewiesen ist. Häufig ist das Verhältnis zwischen Arbeitsspeicher- und vCPU konsistent.
3. Max. temporärer Speicher (GiB) – Der RAM in Gigabyte, der der Instance für nicht persistenten temporären Dateispeicher zugewiesen wird.
4. Max. Bandbreite (Mbit/s) – Die maximale Bandbreite in Megabit pro Sekunde. Dividieren Sie durch 8, um den erwarteten Durchsatz in Megabyte pro Sekunde zu erhalten.
5. Netzwerkleistung – Die Netzwerkgeschwindigkeit im Vergleich zu anderen Instance-Klassen.
6. Unterstützende Engines – Die Amazon DocumentDB-Enignisse, die die Instance-Klasse unterstützen.

Bestimmen des Status einer Instance

Mehr zu den gültigen Instance-Zuständen, ihre Bedeutung und die Bestimmung des Status Ihrer Instances finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Amazon DocumentDB-Instance-Lebenszyklus

Der Lebenszyklus einer Amazon DocumentDB-Instance umfasst das Erstellen, Ändern, Warten und Aktualisieren, das Ausführen von Backups und Wiederherstellungen, das Neustarten und das Löschen der Instance. Dieser Abschnitt enthält Informationen darüber, wie Sie diese Prozesse abschließen können.

Themen

- Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster

- Beschreiben von Amazon DocumentDB-Instances
- Ändern einer Amazon DocumentDB-Instance
- Neustarten einer Amazon DocumentDB-Instance
- Löschen einer Amazon DocumentDB-Instance

Sie können eine neue Amazon DocumentDB-Instance mit der AWS Management Console oder der erstellen AWS CLI. Um eine Instance zu einem Cluster hinzuzufügen, muss sich der Cluster im Zustand available befinden. Zu einem angehaltenen Cluster können keine Instances hinzugefügt werden. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis der Cluster verfügbar wird, und fügen Sie dann eine Instance hinzu. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

> **Note**
>
> Wenn Sie einen Amazon DocumentDB-Cluster mit der Konsole erstellen, wird automatisch gleichzeitig eine Instance für Sie erstellt. Wenn Sie zusätzliche Instances erstellen möchten, verwenden Sie eines der folgenden Verfahren.

Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster

Using the AWS Management Console

Gehen Sie wie folgt vor, um mithilfe der Amazon DocumentDB-Konsole eine Instance für Ihren Cluster zu erstellen.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Um den Cluster zu wählen, dem Sie eine Instance hinzufügen möchten, wählen Sie die Schaltfläche links vom Namen des Clusters aus.
4. Wählen Sie Actions (Aktionen) und dann Add instances (Instances hinzufügen) aus.
5. Wiederholen Sie die folgenden Schritte für jede Instance, die Sie dem Cluster hinzufügen möchten, auf der Seite Add instance to: (Instance hinzufügen zu:)<cluster-name>. Sie können bis zu 15 haben.

   a. Instance-Kennung – Sie können entweder eine eindeutige Kennung für diese Instance eingeben oder Amazon DocumentDB erlauben, die Instance-Kennung basierend auf der Cluster-Kennung bereitzustellen.

      Einschränkungen für Instance-Benennungen:

      - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Instances in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.

   b. Instance-Klasse – Wählen Sie aus der Dropdown-Liste den gewünschten Instance-Typ für diese Instance aus.

   c. Hochstufungsstufe – Wählen Sie in der Dropdown-Liste die Hochstufungsstufe für Ihre Instance aus oder wählen Sie Keine Präferenz, damit Amazon DocumentDB die Hochstufungsstufe für Ihre Instance festlegen kann. Niedrigere Nummern bedeuten eine höhere Priorität. Weitere Informationen finden Sie unter Steuern des Failover-Ziels.

   d. Zum Hinzufügen weiterer Instances wählen Sie die Option Add additional instances (Weitere Instances hinzufügen) und wiederholen Sie die Schritte a, b und c.

6. Schließen Sie den Vorgang ab.

   - Wenn Sie die Instances zu Ihrem Cluster hinzufügen möchten, klicken Sie auf Create (Erstellen).
   - Um die Operation abzubrechen, wählen Sie Abbrechen aus.

Es dauert einige Minuten, bis eine Instance erstellt ist. Sie können die Konsole oder verwenden AWS CLI , um den Status der Instance anzuzeigen. Weitere Informationen finden Sie unter Den Status einer Instanz überwachen.

Using the AWS CLI

Verwenden Sie die `-create-db-instance` AWS CLI Operation mit den folgenden Parametern, um die primäre Instance für Ihren Cluster zu erstellen.

- **`--db-instance-class`** – Erforderlich. Die Rechen- und Speicherkapazität der Instance, beispielsweise `db.m4.large`. Nicht alle Instance-Klassen sind in allen verfügbar AWS-Regionen.
- **`--db-instance-identifier`** – Erforderlich. Eine -Zeichenfolge in Kleinbuchstaben, die die Instance bezeichnet.

  Instance-Benennungseinschränkungen:

  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Instances in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Kontound Region eindeutig sein.
- **`--engine`** – Erforderlich. Der Wert muss `docdb` sein.
- **`--availability-zone`** – Optional. Die Availability Zone, in der diese Instance erstellt werden soll. Verwenden Sie diesen Parameter, um Ihre Instances in verschiedenen Availability Zones zur Erhöhung der Fehlertoleranz zu finden. Weitere Informationen finden Sie unter Amazon DocumentDB Hochverfügbarkeit und -Replikation.
- **`--promotion-tier`** – Optional. Die Failover-Prioritätsstufe für diese Instance. Muss zwischen 0 und 15 liegen, wobei niedrigere Nummern eine höhere Priorität bedeuten. Weitere Informationen finden Sie unter Steuern des Failover-Ziels.

1. Bestimmen Sie zunächst, in welchen Availability Zones Sie Ihre Instance erstellen können.

   Wenn Sie die Availability Zone angeben möchten, bevor Sie Ihre Instance erstellen, führen Sie den folgenden Befehl aus, um festzustellen, welche Availability Zones für Ihren Amazon DocumentDB-Cluster verfügbar sind.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
       --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
       --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-cluster",
        [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ]
    ]
]
```

2. Bestimmen Sie dann, welche Instance-Klassen Sie in Ihrer Region erstellen können.

   Führen Sie den folgenden Befehl aus, um festzustellen, welche Instance-Klassen Ihnen in Ihrer Region zur Verfügung stehen. Wählen Sie aus der Ausgabe eine Instance-Klasse für die Instance aus, die Sie Ihrem Amazon DocumentDB-Cluster hinzufügen möchten.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-orderable-db-instance-options \
           --engine docdb \
           --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
   ```

   Für Windows:

   ```
   aws docdb describe-orderable-db-instance-options ^
           --engine docdb ^
   ```

```
        --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    "db.r5.16xlarge",
    "db.r5.2xlarge",
    "db.r5.4xlarge",
    "db.r5.8xlarge",
    "db.r5.large",
    "db.r5.xlarge"
]
```

3. Fügen Sie Ihrem Amazon DocumentDB-Cluster schließlich eine Instance hinzu.

   Führen Sie den folgenden Befehl aus, um Ihrem Amazon DocumentDB-Cluster eine Instance hinzuzufügen.

   Für Linux, macOS oder Unix:

```
aws docdb create-db-instance \
       --db-cluster-identifier sample-cluster \
       --db-instance-identifier sample-instance-2 \
       --availability-zone us-east-1b \
       --promotion-tier 2 \
       --db-instance-class db.r5.xlarge \
       --engine docdb
```

   Für Windows:

```
aws docdb create-db-instance ^
       --db-cluster-identifier sample-cluster ^
       --db-instance-identifier sample-instance-2 ^
       --availability-zone us-east-1b ^
       --promotion-tier 2 ^
       --db-instance-class db.r5.xlarge ^
       --engine docdb
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
```

```
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance-2",
        "DBInstanceClass": "db.r5.xlarge",
        "Engine": "docdb",
        "DBInstanceStatus": "creating",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1b",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:11:35-sun:12:05",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
```

```
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 2,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2"
    }
}
```

Es dauert einige Minuten, bis die Instance erstellt ist. Sie können die Konsole oder verwenden AWS CLI , um den Status der Instance anzuzeigen. Weitere Informationen finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Beschreiben von Amazon DocumentDB-Instances

Sie können entweder die Amazon DocumentDB-Managementkonsole oder die AWS CLI verwenden, um Details wie Verbindungsendpunkte, Sicherheitsgruppen-VPCs, Zertifizierungsstelle und Parametergruppen zu Ihren Amazon DocumentDB-Instances anzuzeigen.

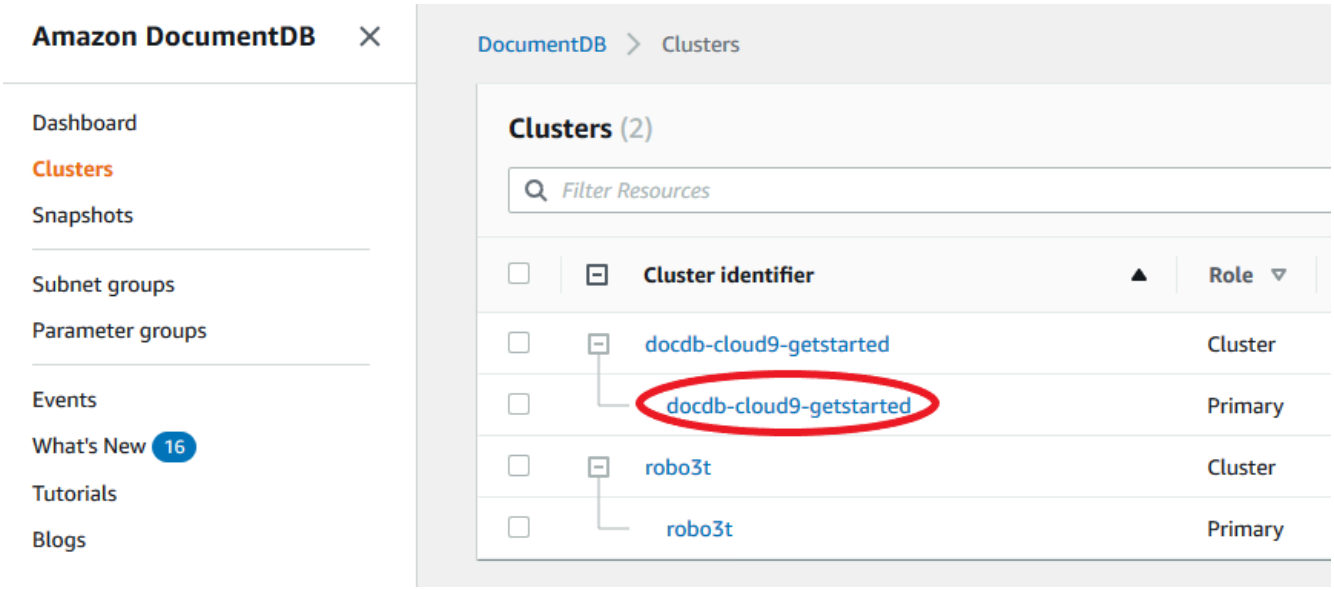
Using the AWS Management Console

Gehen Sie folgendermaßen vor, um die Details Ihrer Instances mithilfe der AWS Management Console anzuzeigen.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Navigationsfeld Cluster sehen Sie die Spalte Cluster-ID . Ihre Instances sind unter Cluster aufgeführt, ähnlich wie im folgenden Screenshot.



4. Wählen Sie in der Liste der Instances den Namen der Instance aus, deren Details Sie anzeigen möchten. Die Informationen über die Instance sind in die folgenden Gruppierungen unterteilt:

   - Zusammenfassung – Allgemeine Informationen über die Instance, einschließlich Engine-Version, Klasse, Status und ausstehender Wartung.
   - Konnektivität und Sicherheit – Im Abschnitt Verbinden werden die Verbindungsendpunkte für die Verbindung mit dieser Instance mit der mongo-Shell oder mit einer Anwendung aufgeführt. Im Abschnitt Sicherheitsgruppen werden die Sicherheitsgruppen aufgeführt, die dieser Instance zugeordnet sind, sowie ihre VPC-ID und Beschreibungen.
   - Konfiguration – Im Abschnitt Details werden die Konfigurationen und der Status der Instance aufgeführt, einschließlich Amazon-Ressourcenname (ARN), Endpunkt, Rolle, Klasse und Zertifizierungsstelle der Instance. Außerdem werden die Sicherheits- und Netzwerkeinstellungen der Instance sowie die Sicherungsinformationen aufgelistet.
     Im Abschnitt Cluster-Details werden die Details des Clusters aufgeführt, zu dem diese Instance gehört. Im Abschnitt Cluster-Instances werden alle Instances aufgeführt, die zu Ihrem Cluster gehören, mit dem Status der einzelnen Instance-Rollen und Cluster-Parametergruppen.

> **Note**
>
> Sie können den Ihrer Instance zugeordneten Cluster ändern, indem Sie neben dem Header Cluster details die Option Modify auswählen. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

- Überwachung – Die CloudWatch Protokollmetriken für diese Instance. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.
- Ereignisse und Tags – Im Abschnitt Aktuelle Ereignisse werden die letzten Ereignisse für diese Instance aufgeführt. Amazon DocumentDB zeichnet Ereignisse auf, die sich auf Ihre Cluster, Instances, Snapshots, Sicherheitsgruppen und Cluster-Parametergruppen beziehen. Zu diesen Informationen gehören Datum, Uhrzeit und Nachricht, die jedem Ereignis zugeordnet sind. Der Abschnitt Tags listet die an diesen Cluster angehängten Tags auf. Weitere Informationen finden Sie unter Taggen von Amazon DocumentDB-Ressourcen.

Using the AWS CLI

Um die Details Ihrer Amazon DocumentDB-Instances mithilfe der anzuzeigen AWS CLI, verwenden Sie den Befehl , `describe-db-clusters` wie in den folgenden Beispielen gezeigt. Weitere Informationen finden Sie unter `DescribeDBInstances` in der API-Referenz für Amazon DocumentDB Resource Management.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen wie Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS geteilt wird. Der `filterName=engine,Values=docdb` Filterparameter gibt nur Amazon DocumentDB-Cluster zurück.

1. Listen Sie alle Amazon DocumentDB-Instances auf.

   Der folgende AWS CLI Code listet die Details für alle Amazon DocumentDB-Instances in einer Region auf.

   Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --filter Name=engine,Values=docdb
```

Für Windows:

```
aws docdb describe-db-instances \
    --filter Name=engine,Values=docdb
```

2. Auflisten aller Details für eine angegebene Amazon DocumentDB-Instance

Der folgende Code listet die Details für `sample-cluster-instance` auf. Das Einschließen des Parameters `--db-instance-identifier` mit dem Namen einer Instance beschränkt die Ausgabe auf Informationen zu dieser bestimmten Instance.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Für Windows:

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

Die Ausgabe dieser Operation sieht folgendermaßen aus.

```
{
    "DBInstances": [
        {
            "DbiResourceId": "db-BJKKB54PIDV5QFKGVRX5T3S6GM",
            "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-
cluster-instance-00",
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-77186e0d",
                    "Status": "active"
                }
            ],
            "DBInstanceClass": "db.r5.large",
            "DBInstanceStatus": "creating",
```

```
            "AutoMinorVersionUpgrade": true,
            "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
            "BackupRetentionPeriod": 1,
            "StorageEncrypted": true,
            "DBClusterIdentifier": "sample-cluster",
            "EngineVersion": "3.6.0",
            "AvailabilityZone": "us-east-1a",
            "Engine": "docdb",
            "PromotionTier": 2,
            "DBInstanceIdentifier": "sample-cluster-instance",
            "PreferredBackupWindow": "00:00-00:30",
            "PubliclyAccessible": false,
            "DBSubnetGroup": {
                "DBSubnetGroupName": "default",
                "Subnets": [
                    {
                        "SubnetIdentifier": "subnet-4e26d263",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1a"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-afc329f4",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1c"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-b3806e8f",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1e"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-53ab3636",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1d"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
```

```
                        "SubnetIdentifier": "subnet-991cb8d0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-29ab1025",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1f"
                        },
                        "SubnetStatus": "Active"
                    }
                ],
                "VpcId": "vpc-91280df6",
                "DBSubnetGroupDescription": "default",
                "SubnetGroupStatus": "Complete"
            },
            "PendingModifiedValues": {},
            "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-
a50b-44d4-b6a0-a177d5ff730b"
        }
    ]
}
```

Ändern einer Amazon DocumentDB-Instance

Sie können Ihre Amazon DocumentDB-Instance entweder über die AWS Management Console oder die ändern AWS CLI. Nur Instances im Status available können geändert werden. Eine angehaltene Instance kann nicht geändert werden. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis die Instance verfügbar wird, und nehmen Sie dann die gewünschten Änderungen vor. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

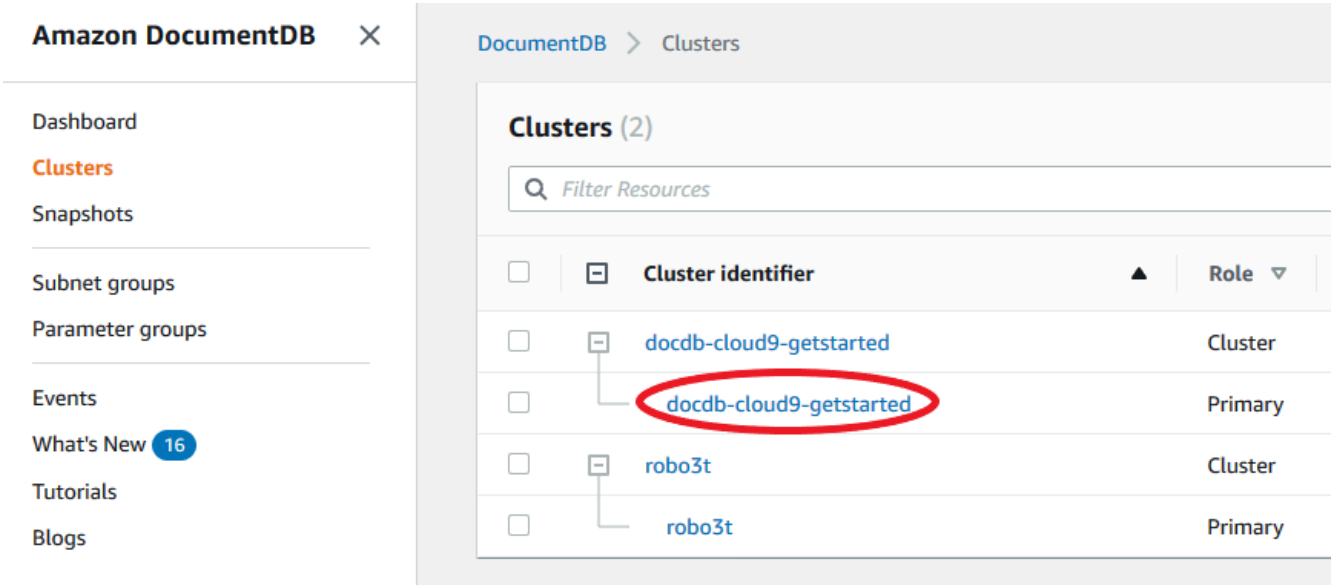
Using the AWS Management Console

Führen Sie die folgenden Schritte aus, um eine bestimmte Amazon DocumentDB-Instance mithilfe der Konsole zu ändern.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Navigationsfeld Cluster sehen Sie die Spalte Cluster-ID . Ihre Instances sind unter Cluster aufgeführt, ähnlich wie im folgenden Screenshot.



4. Aktivieren Sie das Kontrollkästchen links neben der Instance, die Sie ändern möchten.
5. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
6. Nehmen Sie im Bereich Modify instance: <instance-name> (Instance ändern: <instance-name>) die gewünschten Änderungen vor. Sie können die folgenden Änderungen ausführen:

   - Instance-Spezifikationen – Die Instance-ID und -Klasse. Benennungseinschränkungen für Instance-IDs:
     - Instance-Kennung – Geben Sie einen Namen ein, der für alle Instances, die Ihrem AWS-Konto in der aktuellen Region gehören, eindeutig ist. Die Instance-ID muss [1–63] alphanumerische Zeichen oder Bindestriche enthalten, einen Buchstaben

als erstes Zeichen haben und darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

- Instance-Klasse – Wählen Sie im Dropdown-Menü eine Instance-Klasse für Ihre Amazon DocumentDB-Instance aus. Weitere Informationen finden Sie unter Verwalten von Instance-Klassen.
- Zertifizierungsstelle – Serverzertifikat für diese Instance. Weitere Informationen finden Sie unter Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate.
- Failover – Während des Failovers wird die Instance mit der höchsten Hochstufungsstufe zur primären Instance hochgestuft. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Failover.
- Wartung – Das Wartungsfenster, in dem ausstehende Änderungen oder Patches auf Instances im Cluster angewendet werden.

7. Wenn Sie fertig sind, wählen Sie Continue (Weiter) aus, um eine Zusammenfassung Ihrer Änderungen anzuzeigen.
8. Nachdem Sie Ihre Änderungen überprüft haben, können Sie diese sofort oder während des nächsten Wartungsfensters unter Scheduling of modifications (Planen von Änderungen) anwenden. Klicken Sie auf Modify instance (Instance ändern), um Ihre Änderungen zu speichern. Alternativ können Sie auf Cancel (Abbrechen) klicken, um Ihre Änderungen zu verwerfen.

Es dauert einige Minuten, bis Ihre Änderungen übernommen werden. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Using the AWS CLI

Um eine bestimmte Amazon DocumentDB-Instance mithilfe der zu ändern AWS CLI, verwenden Sie die `modify-db-instance` mit den folgenden Parametern. Weitere Informationen finden Sie unter ModifyDBInstance. Der folgende Code ändert die Instance-Klasse für die Instance `sample-instance` in `db.r5.large`.

Parameter

- **`--db-instance-identifier`** – Erforderlich. Die Kennung der zu ändernden Instance.
- **`--db-instance-class`** – Optional. Die neue Rechen- und Speicherkapazität der Instance, z. B. `db.r5.large`. Nicht alle Instance-Klassen sind in allen verfügbar AWS-Regionen.

Wenn Sie die Instance-Klasse ändern, kommt es während der Änderung zu einem Ausfall. Die Änderung wird während des nächsten Wartungsfensters angewendet, es sei denn, für diese Anforderung `ApplyImmediately` wird als wahr angegeben.

- **`--apply-immediately`** oder **`--no-apply-immediately`** – Optional. Gibt an, ob diese Änderung sofort angewandt oder bis zum nächsten Wartungszeitraum verschoben werden sollte. Wenn dieser Parameter weggelassen wird, wird die Änderung im nächsten Wartungszeitraum ausgeführt.

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-instance \
       --db-instance-identifier sample-instance \
       --db-instance-class db.r5.large \
       --apply-immediately
```

Für Windows:

```
aws docdb modify-db-instance ^
       --db-instance-identifier sample-instance ^
       --db-instance-class db.r5.large ^
       --apply-immediately
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "sample-instance-1",
            "DBInstanceClass": "db.r5.large",
            "Engine": "docdb",
            "DBInstanceStatus": "modifying",
            "Endpoint": {
                "Address": "sample-instance-1.node.us-east-1.docdb.amazonaws.com",
                "Port": 27017,
                "HostedZoneId": "ABCDEFGHIJKLM"
            },
            "InstanceCreateTime": "2020-01-10T22:18:55.921Z",
            "PreferredBackupWindow": "02:00-02:30",
            "BackupRetentionPeriod": 1,
```

```
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-abcd0123",
                    "Status": "active"
                }
            ],
            "AvailabilityZone": "us-east-1a",
            "DBSubnetGroup": {
                "DBSubnetGroupName": "default",
                "DBSubnetGroupDescription": "default",
                "VpcId": "vpc-abcd0123",
                "SubnetGroupStatus": "Complete",
                "Subnets": [
                    {
                        "SubnetIdentifier": "subnet-abcd0123",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1a"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-abcd0123",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    }
                ]
            },
            "PreferredMaintenanceWindow": "sun:10:57-sun:11:27",
            "PendingModifiedValues": {
                "DBInstanceClass": "db.r5.large"
            },
            "EngineVersion": "3.6.0",
            "AutoMinorVersionUpgrade": true,
            "PubliclyAccessible": false,
            "DBClusterIdentifier": "sample-cluster",
            "StorageEncrypted": true,
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY",
            "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
            "CACertificateIdentifier": "rds-ca-2019",
            "PromotionTier": 1,
```

```
            "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:sample-
instance-1",
            "EnabledCloudwatchLogsExports": [
                "profiler"
            ]
        }
    ]
}
```

Es dauert einige Minuten, bis Ihre Änderungen angewendet werden. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können den Status der Instance mit der AWS Management Console oder der überwachen AWS CLI. Weitere Informationen finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Neustarten einer Amazon DocumentDB-Instance

Gelegentlich müssen Sie Ihre Amazon DocumentDB-Instance möglicherweise neu starten, normalerweise aus Wartungsgründen. Wenn Sie bestimmte Änderungen vornehmen, z. B. die Clusterparametergruppe, die einem Cluster zugeordnet ist, müssen Sie die Instances im Cluster neu starten, damit die Änderungen wirksam werden. Sie können eine angegebene Instance mithilfe der AWS Management Console oder der neu starten AWS CLI.

Durch den Neustart einer Instance wird der Datenbank-Engine-Service neu gestartet. Das Neustarten bewirkt einen vorübergehenden Nutzungsausfall, wobei der Status für die Instance währenddessen auf `rebooting` gesetzt wird. Ein Amazon DocumentDB-Ereignis wird erstellt, wenn der Neustart abgeschlossen ist.

Ein Neustart einer Instance führt nicht zu einem Failover. Um ein Failover für einen Amazon DocumentDB-Cluster durchzuführen, verwenden Sie die AWS CLI Operation AWS Management Console oder `failover-db-cluster`. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Failover.

Sie können Ihre Instance nicht neu starten, wenn sie sich nicht im verfügbaren Zustand befindet. Ihre Datenbank kann aus mehreren Gründen nicht verfügbar sein, zum Beispiel aufgrund einer zuvor angeforderten Änderung oder einer Aktion im Wartungsfenster. Weitere Informationen zu Instance-Status finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

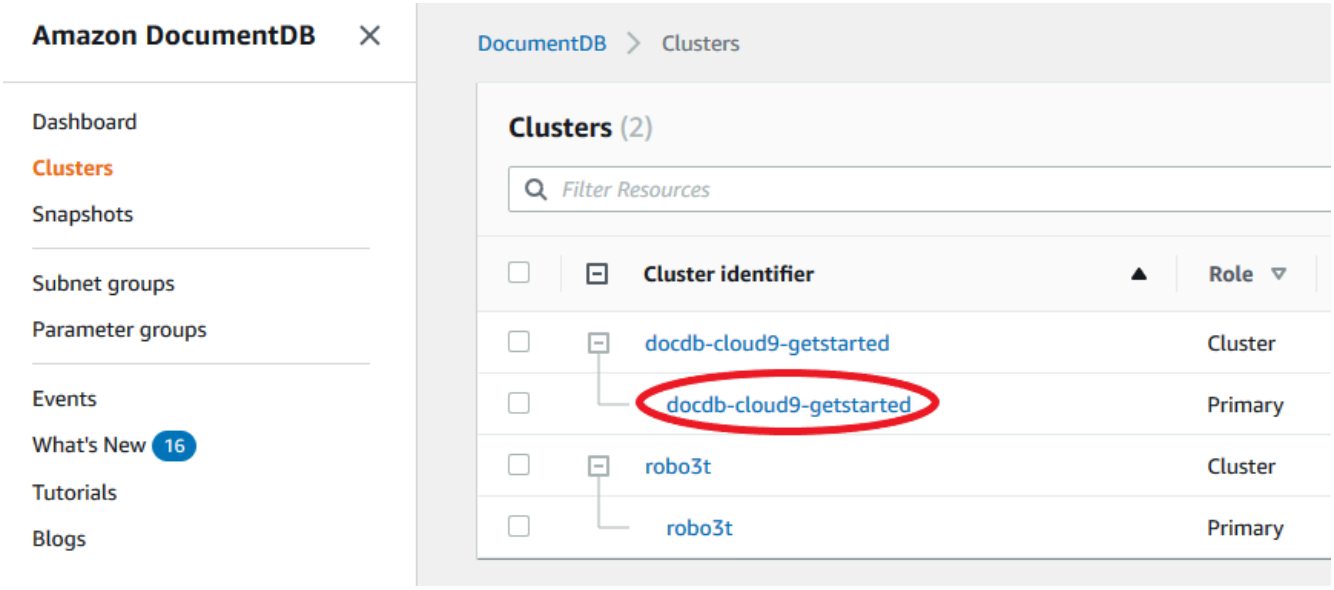
Using the AWS Management Console

Das folgende Verfahren startet eine Instance neu, die Sie unter Verwendung der Konsole angeben.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Navigationsfeld Cluster sehen Sie die Spalte Cluster-ID . Ihre Instances sind unter Cluster aufgeführt, ähnlich wie im folgenden Screenshot.



4. Aktivieren Sie das Kontrollkästchen links neben der Instance, die Sie neu starten möchten.
5. Wählen Sie Actions (Aktionen), Reboot (Neustart) und dann Reboot (Neustart)aus, um den Neustart zu bestätigen.

Es dauert einige Minuten, bis Ihre Instance neu gestartet wird. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den

Status der Instance überwachen. Weitere Informationen finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Using the AWS CLI

Um eine Amazon DocumentDB-Instance neu zu starten, verwenden Sie die `-reboot-db-instance`Operation mit dem `---db-instance-identifier`Parameter. Dieser Parameter gibt den Bezeichner für die Instance an, die neu gestartet werden soll.

Der folgende Code startet die Instance `sample-instance` neu.

Example

Für Linux, macOS oder Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-instance
```

Für Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-instance
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "rebooting",
        "Endpoint": {
            "Address": "sample-instance.node.us-east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
```

```
            }
        ],
        "AvailabilityZone": "us-east-1c",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-abcd0123",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ]
    }
}
```

Es dauert einige Minuten, bis Ihre Instance neu gestartet wird. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Löschen einer Amazon DocumentDB-Instance

Sie können Ihre Amazon DocumentDB-Instance entweder über die AWS Management Console oder die löschen AWS CLI. Nur Instances im Status available können gelöscht werden. Eine angehaltene Instance kann nicht gelöscht werden. Wenn der Amazon DocumentDB-Cluster, der Ihre Instance enthält, gestoppt ist, starten Sie zuerst den Cluster, warten Sie, bis die Instance verfügbar ist, und löschen Sie dann die Instance. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

> **Note**
>
> Amazon DocumentDB speichert alle Ihre Daten im Cluster-Volume. Die Daten bleiben in diesem Cluster-Volume erhalten. Dies gilt auch dann, wenn Sie alle Instances von Ihrem Cluster entfernen. Wenn Sie erneut auf die Daten zugreifen müssen, können Sie jederzeit dem Cluster eine Instance hinzufügen und da fortfahren, wo Sie aufgehört haben.

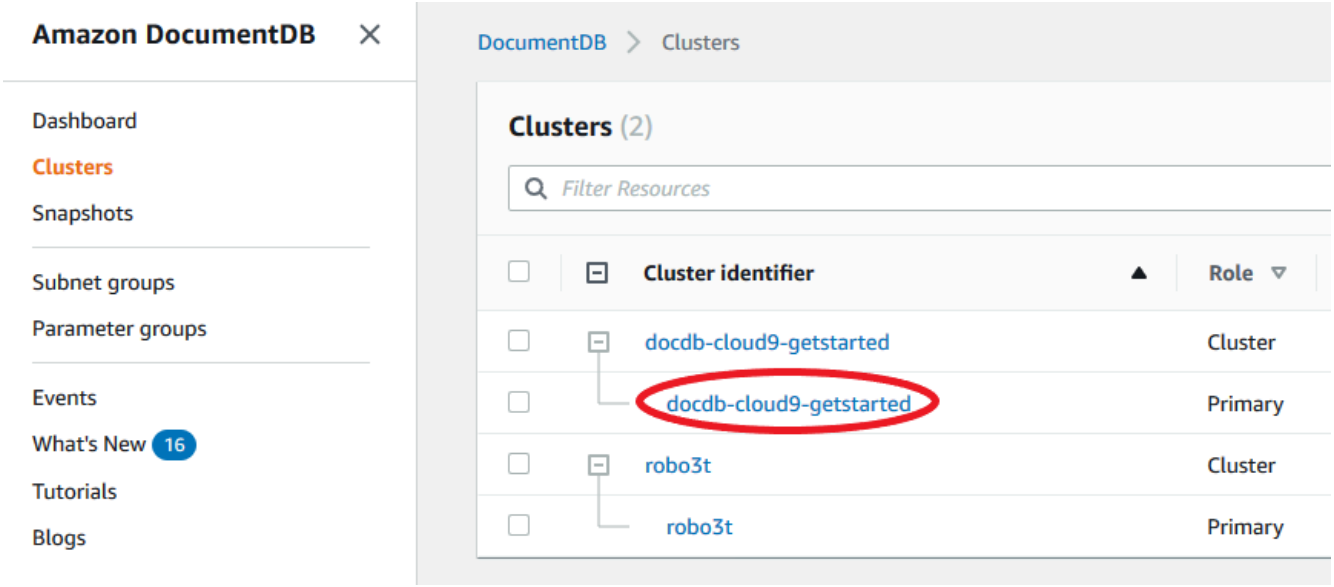
Using the AWS Management Console

Mit dem folgenden Verfahren wird eine angegebene Amazon DocumentDB-Instance mithilfe der Konsole gelöscht.

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Navigationsfeld Cluster sehen Sie die Spalte Cluster-ID . Ihre Instances sind unter Cluster aufgeführt, ähnlich wie im folgenden Screenshot.



4. Aktivieren Sie das Kontrollkästchen links neben der Instance, die Sie löschen möchten.
5. Wählen Sie Actions (Aktionen) und anschließend Delete (Löschen) aus.

   1. Wenn Sie die letzte Instance in Ihrem Cluster löschen:

      - Create final cluster snapshot? (Endgültigen Cluster-Snapshot erstellen?) – Wählen Sie Ja, wenn Sie einen endgültigen Snapshot erstellen möchten, bevor der Cluster gelöscht wird. Wählen Sie andernfalls No (Nein) aus.
      - Endgültiger Snapshot-Name – Wenn Sie einen endgültigen Snapshot erstellen möchten, geben Sie die Cluster-Snapshot-ID des neu erstellten Cluster-Snapshots ein.
      - Delete <instance-name> instance? (Instance <instance-name> löschen?) – Geben Sie den Ausdruck Löschen des gesamten Clusters in das Feld ein, um den Löschvorgang zu bestätigen.

   2. Wenn Sie nicht die letzte Instance in Ihrem Cluster löschen:

      - Delete <instance-name> instance? (Instance <instance-name> löschen?) – Geben Sie den Ausdruck Löschen in das Feld ein, um den Löschvorgang zu bestätigen.

6. Wählen Sie Delete (Löschen), um die Instance zu löschen.

Es dauert einige Minuten, bis die Instance gelöscht ist. Informationen zum Überwachen des Status einer Instance finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

Using the AWS CLI

Mit dem folgenden Verfahren wird eine Amazon DocumentDB-Instance mithilfe der gelöscht AWS CLI.

1. Bestimmen Sie zunächst, wie viele Instances sich in Ihrem Amazon DocumentDB-Cluster befinden:

   Führen Sie den `describe-db-clusters`-Befehl wie folgt aus, um festzustellen, wie viele Instances sich in Ihrem Cluster befinden.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].
   [DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

   ```
   [
       [
           "sample-cluster",
           [
               "sample-instance-1",
               "sample-instance-2"
           ]
       ]
   ]
   ```

2. Wenn es mehr als eine Instance in Ihrem Amazon DocumentDB-Cluster gibt:

   Um eine angegebene Amazon DocumentDB-Instance zu löschen, verwenden Sie den `-delete-db-instance`Befehl mit dem `---db-instance-identifier`Parameter, wie unten gezeigt. Es dauert einige Minuten, bis die Instance gelöscht ist. Informationen zum Überwachen des Status einer Instance finden Sie unter Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance.

   ```
   aws docdb delete-db-instance \
          --db-instance-identifier sample-instance-2
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance-2",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "deleting",
        "Endpoint": {
            "Address": "sample-instance-2.node.us-east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1c",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
```

```
        "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ]
    }
}
```

3. Wenn die Instance, die Sie löschen möchten, die letzte Instance in Ihrem Amazon DocumentDB-Cluster ist:

   Wenn Sie die letzte Instance in einem Amazon DocumentDB-Cluster löschen, löschen Sie auch diesen Cluster und die automatischen Snapshots und kontinuierlichen Backups, die diesem Cluster zugeordnet sind.

   Um die letzte Instance in Ihrem Cluster zu löschen, können Sie den Cluster löschen und optional einen endgültigen Snapshot erstellen. Weitere Informationen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Löschschutz

Durch das Löschen der letzten Instance eines Amazon DocumentDB-Clusters werden auch der Cluster sowie die automatischen Snapshots und kontinuierlichen Backups gelöscht, die diesem Cluster zugeordnet sind. Amazon DocumentDB erzwingt den Löschschutz für einen Cluster, unabhängig davon, ob Sie den Löschvorgang mit der AWS Management Console oder der durchführen AWS CLI. Sie können einen Cluster nicht löschen, solange der Löschschutz aktiviert ist.

Um einen Cluster mit aktiviertem Löschschutz zu löschen, müssen Sie zuerst den Cluster ändern und den Löschschutz deaktivieren. Weitere Informationen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Amazon DocumentDB-Subnetzgruppen verwalten

Eine Virtual Private Cloud (VPC) ist ein virtuelles Netzwerk für Ihre BedürfnisseAWS-Konto. Es ist von anderen virtuellen Netzwerken in der AWS Cloud getrennt. Sie können IhreAWS -Ressourcen, z. B. Amazon DocumentDB, in Ihrer Amazon-VPC VPC können. Sie können einen IP-Adressbereich für die VPC angeben, Subnetze hinzufügen, Sicherheitsgruppen zuordnen und Routing-Tabellen konfigurieren.

Ein Subnetz ist ein Bereich an IP-Adressen in Ihrer Amazon-VPC. Sie können AWS-Ressourcen in einem von Ihnen angegebenen Subnetz starten. Verwenden Sie öffentliche Subnetze für Ressourcen, die mit dem Internet verbunden sein müssen. Verwenden Sie private Subnetze für Ressourcen, die nicht mit dem Internet verbunden sein werden. Weitere Informationen zu öffentlichen und privaten Subnetzen finden Sie unter VPC-Grundlagen im Amazon Virtual Private Cloud-Benutzerhandbuch unter VPC-Grundlagen im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Eine DB-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie in einer VPC erstellen, welche Sie dann für Ihre Cluster festlegen. Mithilfe einer Subnetzgruppe können Sie beim Erstellen von Clustern eine bestimmte VPC festlegen. Wenn Sie die `default` Subnetzgruppe verwenden, umfasst diese alle Subnetze in der VPC.

Jede DB-Subnetzgruppe sollte über Subnetze in mindestens zwei Availability Zones in einer bestimmten -Region verfügen. Wenn einen DB-Cluster in einer VPC erstellen, müssen Sie eine DB-Subnetzgruppe auswählen. Amazon DocumentDB verwendet diese DB-Subnetzgruppe und Ihre bevorzugte Availability Zone, um ein Subnetz und eine IP-Adresse innerhalb dieses Subnetzes auszuwählen, die mit Ihrem Cluster verknüpft werden sollen. Wenn die primäre Instance ausfällt, kann Amazon DocumentDB eine entsprechende Replica zur neuen primären Instance hochstufen. Anschließend kann mithilfe einer IP-Adresse aus dem Subnetz, in dem sich der vorherige primäre Cluster befand, eine neue Replikat-Instance erstellt werden.

Wenn Amazon DocumentDB eine Instance in einer VPC erstellt, wird Ihrem Cluster mithilfe einer aus Ihrer DB-Subnetzgruppe eine Netzwerkschnittstelle zugewiesen. Es wird ausdrücklich empfohlen, den DNS-Namen zu verwenden, da sich die zugrunde liegende IP-Adresse ändern kann (z. B während eines Failovers). Weitere Informationen finden Sie unter Amazon DocumentDB-Endpunkte.

Informationen zum Erstellen eigener VPC und Subnetze finden Sie unter Arbeiten mit VPCs und Subnetzen im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Themen

Erstellen einer Amazon DocumentDB Documente-Gruppe

Wenn Sie einen Amazon DocumentDB-Cluster erstellen, müssen Sie eine Amazon VPC und die entsprechende Subnetzgruppe innerhalb dieser Amazon VPC auswählen, um Ihren Cluster zu starten. Subnetze bestimmen die Availability Zone und den IP-Bereich innerhalb der Availability Zone, die Sie zum Starten einer Instance verwenden möchten.

Eine Subnetzgruppe ist ein benannter Satz von Subnetzen (oder AZs), mit dem Sie die Availability Zones angeben können, die Sie für den Start von Amazon DocumentDB DocumentDB-Instances verwenden möchten. In einem Cluster mit drei Instances wird beispielsweise empfohlen, dass jede dieser Instanzen in separaten AZs bereitgestellt wird. Dadurch wird die Hochverfügbarkeit optimiert. Wenn also eine einzelne AZ ausfällt, wirkt sich dies nur auf eine einzelne Instanz aus.

Derzeit können Amazon DocumentDB DocumentDB-Instances in bis zu drei AZs bereitgestellt werden. Selbst wenn eine Subnetzgruppe mehr als drei Subnetze hat, können Sie nur drei dieser Subnetze verwenden, um einen Amazon DocumentDB-Cluster zu erstellen. Daher empfehlen wir, dass Sie beim Erstellen einer Subnetzgruppe nur die drei Subnetze auswählen, von denen Sie Ihre Instances bereitstellen möchten.

Zum Beispiel: Ein Cluster wird erstellt und Amazon DocumentDB wählt AZs {1A, 1B und 1C}. Wenn Sie versuchen, eine Instance in AZ {1D} zu erstellen, schlägt der API-Aufruf fehl. Wenn Sie sich jedoch dafür entscheiden, eine Instance zu erstellen, ohne die bestimmte AZ anzugeben, wählt Amazon DocumentDB in Ihrem Namen eine AZ aus. Amazon DocumentDB verwendet einen Algorithmus für den Lastenausgleich der Instances zwischen AZs, um Ihnen zu helfen, eine hohe Verfügbarkeit zu erreichen. Wenn drei Instances bereitgestellt werden, werden sie standardmäßig über drei AZs bereitgestellt und nicht alle in einer einzigen AZ bereitgestellt.

Bewährte Methoden

- Wenn Sie keinen speziellen Grund haben, legen Sie immer eine Subnetzgruppe mit drei Subnetzen an. Dadurch wird sichergestellt, dass Cluster mit drei oder mehr Instances eine höhere Verfügbarkeit erreichen können, da die Instances auf drei AZs bereitgestellt werden.

- Verteilen Sie Instances immer über mehrere AZs, um eine hohe Verfügbarkeit zu erreichen. Platzieren Sie niemals alle Instances für einen Cluster in einer einzigen AZ.
- Da Failover-Ereignisse jederzeit auftreten können, sollten Sie nicht davon ausgehen, dass sich eine primäre Instance oder Replikat-Instance immer in einer bestimmten AZ befinden.

So erstellen Sie eine Subnetzgruppe

Sie können dasAWS Management Console oder verwendenAWS CLI, um eine Amazon DocumentDB-Subnetzgruppe zu erstellen:

Using the AWS Management Console

Führen Sie die folgenden Schritte aus, um eine Amazon DocumentDB Documente-Gruppe zu erstellen.

So erstellen Sie eine Amazon DocumentDB Documente-Gruppe

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon-DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnet groups (Subnetzgruppen) und anschließend Create (Erstellen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Auf der Seite Create subnet group (Subnetzgruppe erstellen):

   a. Im Abschnitt Subnet group details (Subnetzgruppen-Details:

      i. Name — Geben Sie einen aussagekräftigen Namen für die Subnetzgruppe für die Subnetzgruppe für die Subnetzgruppe

      ii. Beschreibung – Geben Sie eine Beschreibung für die Subnetzgruppe ein.

   b. Im Abschnitt Add subnets (Subnetze hinzufügen):

      i. VPC —Wählen Sie in der Liste eine VPC für diese Subnetzgruppe aus.

ii. Führen Sie eine der folgenden Aktionen aus:

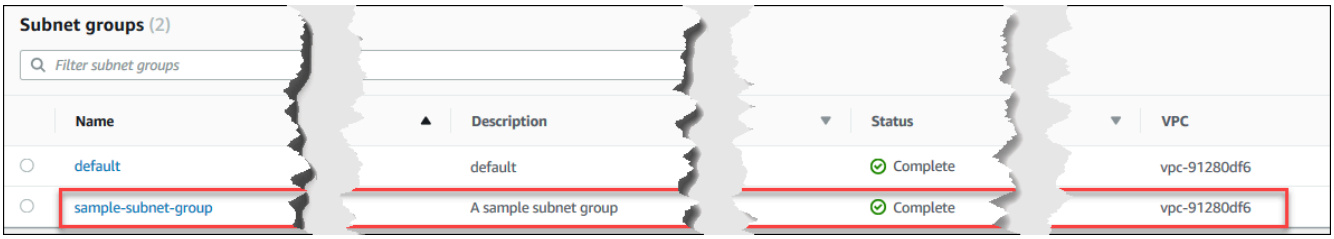
- Um alle Subnetze in der ausgewählten VPC miteinzuschließen, wählen Sie Add all the subnets related to this VPC (Alle zu dieser VPC gehörenden Subnetze hinzufügen).
- Um Subnetze für diese Subnetzgruppe festzulegen, führen Sie folgende Aufgaben für jede Availability Zone aus, für die Sie Subnetze miteinschließen möchten. Sie müssen mindestens zwei Availability Zones einschließen.

  A. Availability Zone —Wählen Sie in der Liste eine Availability Zone aus.

  B. Subnetz —Wählen Sie in der Liste ein Subnetz aus der ausgewählten Availability Zone für diese Subnetzgruppe aus.

  C. Wählen Sie Add subnet (Subnetz hinzufügen) aus.

4. Wählen Sie Create (Erstellen) aus. Wenn die Subnetzgruppe erstellt wurde, wird sie mit Ihren anderen Subnetzgruppen aufgeführt.



Using the AWS CLI

Bevor Sie mit der AWS CLI eine Subnetzgruppe erstellen können, müssen Sie zunächst ermitteln, welche Subnetze verfügbar sind Führen Sie die folgenden AWS CLI-Operation aus, um eine Liste der Availability Zones und ihrer Subnetze anzuzeigen.

Parameter:

- **`--db-subnet-group`**—Fakultativ. Bei Angabe einer bestimmten Subnetzgruppe werden die Availability Zones und Subnetze für diese Gruppe aufgelistet. Durch Auslassen dieses Parameters werden Availability Zones und Subnetze für alle Ihre Subnetzgruppen aufgelistet. Bei Angabe der `default`-Subnetzgruppe werden alle Subnetze der VPC aufgelistet.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default \
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

Für Windows:

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name default ^
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "default",
        [
            [
                "us-east-1a",
                "subnet-4e26d263"
            ],
            [
                "us-east-1c",
                "subnet-afc329f4"
            ],
            [
                "us-east-1e",
                "subnet-b3806e8f"
            ],
            [
                "us-east-1d",
                "subnet-53ab3636"
            ],
            [
                "us-east-1b",
                "subnet-991cb8d0"
            ],
            [
                "us-east-1f",
                "subnet-29ab1025"
            ]
```

```
        ]
    ]
]
```

Sie können eine neue Subnetzgruppe erstellen, indem Sie die Ausgabe des vorherigen Vorgangs verwenden. Die neue Subnetzgruppe muss über Subnetze aus mindestens zwei Availability Zones verfügen.

Parameter:

- **`--db-subnet-group-name`**—Erforderlich. Der Name für diese Subnetzgruppe.
- **`--db-subnet-group-description`**—Erforderlich. Beschreibung dieser Subnetzgruppe.
- **`--subnet-ids`**—Erforderlich. Eine Liste der Subnetze in dieser Subnetzgruppe. Beispiel: `subnet-53ab3636`.
- --Tags —Optional. Eine Liste der Tags (Schlüssel-Wert-Paare) zum Anfügen an diese Subnetzgruppe.

Der folgende Code erstellt die Subnetzgruppe `sample-subnet-group` mit drei Subnetzen, `subnet-4e26d263`, `subnet-afc329f4` und `subnet-b3806e8f`.

Für Linux, macOS oder Unix:

```
aws docdb create-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --db-subnet-group-description "A sample subnet group" \
    --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f \
    --tags Key=tag1,Value=One Key=tag2,Value=2
```

Für Windows:

```
aws docdb create-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group ^
    --db-subnet-group-description "A sample subnet group" ^
    --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f ^
    --tags Key=tag1,Value=One Key=tag2,Value=2
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
```

```
    "DBSubnetGroup": {
        "DBSubnetGroupDescription": "A sample subnet group",
        "DBSubnetGroupName": "sample-subnet-group",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetIdentifier": "subnet-4e26d263",
                "SubnetStatus": "Active"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetIdentifier": "subnet-afc329f4",
                "SubnetStatus": "Active"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetIdentifier": "subnet-b3806e8f",
                "SubnetStatus": "Active"
            }
        ],
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "SubnetGroupStatus": "Complete"
    }
}
```

Beschreibung einer Amazon DocumentDB Documente-Gruppe

Sie können dasAWS Management Console oder das verwendenAWS CLI, um die Details einer Amazon DocumentDB-Subnetzgruppe abzurufen.

Using the AWS Management Console

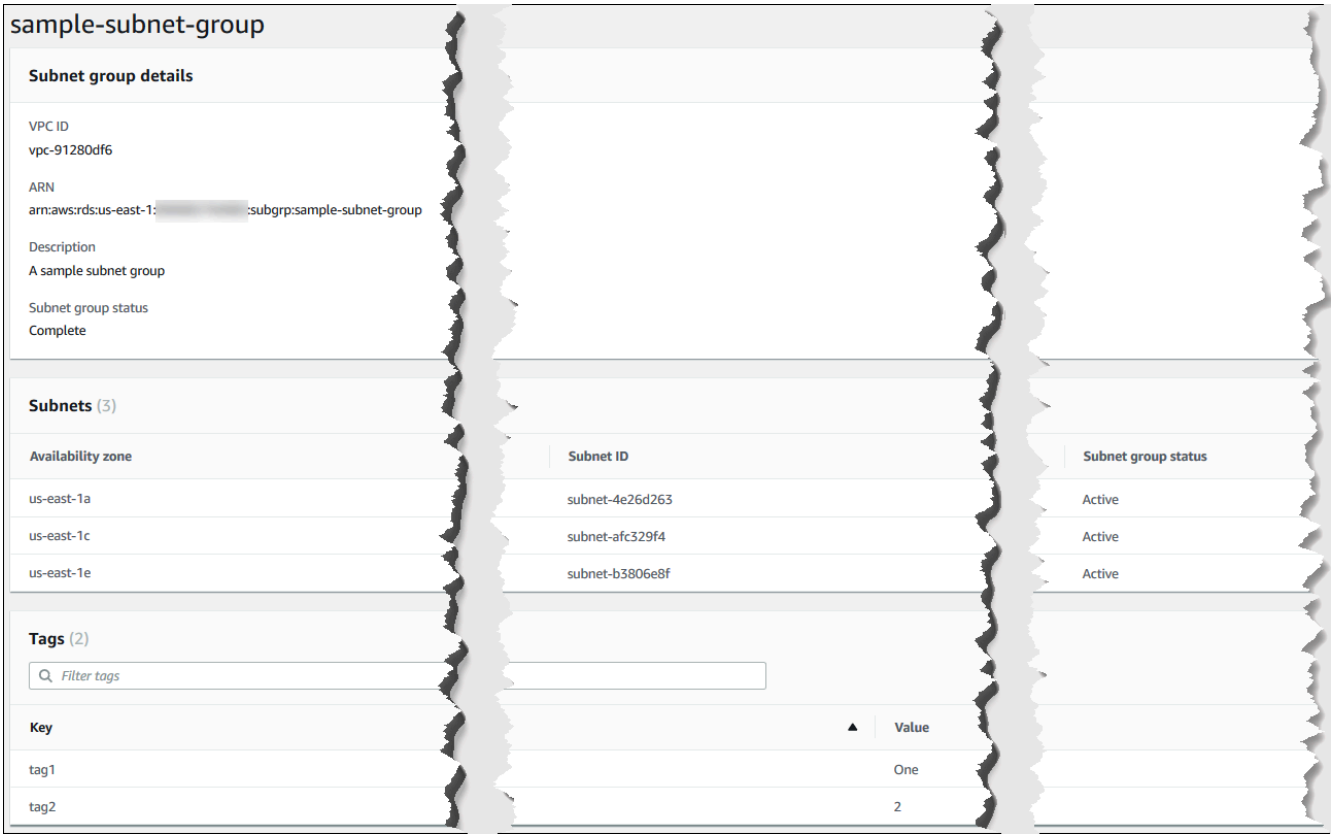
Das folgende Verfahren zeigt Ihnen, wie Sie die Details einer Amazon DocumentDB-Subnetzgruppe abrufen können.

So suchen Sie die Details einer Subnetzgruppe

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon-DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Um die Details einer Subnetzgruppe anzuzeigen, wählen Sie den Namen der Subnetzgruppe.



Using the AWS CLI

Verwenden Sie den`describe-db-subnet-groups` Vorgang mit dem folgenden Parameter, um die Details einer Amazon DocumentDB-Subnetzgruppe zu finden.

Parameter

- `--db-subnet=group-name`—Fakultativ. Falls angegeben, werden Details für die benannte Subnetzgruppe aufgelistet. Falls nicht angegeben, werden Details für bis zu 100 Subnetzgruppen aufgelistet.

Example

Der folgende Code listet die Details für die Subnetzgruppe `sample-subnet-group` auf, die im Abschnitt Erstellen einer Amazon DocumentDB Documente-Gruppe erstellt wurde.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name sample-subnet-group
```

Für Windows:

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name sample-subnet-group
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "VpcId": "vpc-91280df6",
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupName": "sample-subnet-group",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-4e26d263"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
```

```
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-afc329f4"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-b3806e8f"
            }
        ],
        "DBSubnetGroupDescription": "A sample subnet group"
    }
}
```

Ändern einer Amazon DocumentDB Documente-Gruppe

Sie können dasAWS Management Console oder verwendenAWS CLI, um die Beschreibung einer Subnetzgruppe zu ändern oder Subnetze zu einer Amazon DocumentDB-Subnetzgruppe hinzuzufügen oder zu entfernen. Sie können die `default`-Subnetzgruppe jedoch nicht ändern.

Using the AWS Management Console

Sie können die AWS Management Console verwenden, um die Beschreibung eines Subnetzes zu ändern oder um Subnetze hinzuzufügen oder zu entfernen. Denken Sie daran, dass beim Abschluss des Vorgangs mindestens zwei Availability Zones mit Ihrer Subnetzgruppe verbunden sein müssen.

So ändern Sie Ihre Subnetzgruppe

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon-DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus. Klicken Sie dann auf die Schaltfläche links neben dem Namen der Subnetzgruppe. Denken Sie daran, dass Sie die `default`-Subnetzgruppe nicht ändern können.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol

(≡)
aus.

3. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
4. Beschreibung — Um die Beschreibung Ihrer Subnetzgruppe zu ändern, geben Sie eine neue Beschreibung ein.
5. Wenn Sie die mit Ihrer Subnetzgruppe verbundenen Subnetze ändern möchten, führen Sie im Abschnitt Add subnets (Subnetze hinzufügen) eine oder mehrere der folgenden Aktionen durch:
   - Um alle Subnetze aus dieser Subnetzgruppe zu entfernen, wählen Sie Remove all (Alle entfernen) aus.
   - Wenn Sie bestimmte Subnetze aus dieser Subnetzgruppe entfernen möchten, wählen Sie für jedes einzelne Subnetz, das Sie entfernen möchten, Remove (Entfernen) aus.
   - Wenn Sie alle mit dieser VPC verbundenen Subnetze hinzufügen möchten, wählen Sie Add all the subnets related to this VPC (Alle zu dieser VPC gehörenden Subnetze hinzufügen) aus.
   - Wenn Sie bestimmte Subnetze zu dieser Subnetzgruppe hinzufügen möchten, führen Sie die folgenden Aufgaben für jede Availability Zone aus, der Sie ein Subnetz hinzufügen möchten.

     a. Availability Zone —Wählen Sie in der Liste eine neue Availability Zone aus.
     b. Subnetz —Wählen Sie in der Liste ein Subnetz aus der ausgewählten Availability Zone für diese Subnetzgruppe aus.
     c. Wählen Sie Add subnet (Subnetz hinzufügen) aus.
6. Im Bestätigungsdialogfeld:
   - Wählen Sie Modify (Ändern), um diese Änderungen an der Subnetzgruppe vorzunehmen.
   - Wenn die Subnetzgruppe unverändert bleiben soll, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Sie können die AWS CLI verwenden, um die Beschreibung eines Subnetzes zu ändern oder um Subnetze hinzuzufügen oder zu entfernen. Denken Sie daran, dass beim Abschluss des Vorgangs mindestens zwei Availability Zones mit Ihrer Subnetzgruppe verbunden sein müssen. Es ist nicht möglich, die `default`- Subnetzgruppe zu ändern.

Parameter:

- `--db-subnet-group-name`—Erforderlich. Der Name der Amazon DocumentDB-Subnetzgruppe, die Sie ändern.
- `--subnet-ids`—Erforderlich. Eine Liste aller Subnetze, die nach dieser Änderung Teil der Subnetzgruppe sein sollen.

  > **Important**
  >
  > Alle Subnetze, die sich derzeit in der Subnetzgruppe befinden und die nicht in dieser Liste aufgeführt sind, werden aus der Subnetzgruppe entfernt. Wenn Sie Subnetze, die sich derzeit in der Subnetzgruppe befinden, beibehalten möchten, müssen Sie diese in die Liste aufnehmen.

- `--db-subnet-group-description`—Fakultativ. Beschreibung der Subnetzgruppe.

Example

Der folgende Code ändert die Beschreibung und ersetzt die vorhandenen Subnetze durch die Subnetze `subnet-991cb8d0`, `subnet-53ab3636` und `subnet-29ab1025`.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 \
    --db-subnet-group-description "Modified subnet group"
```

Für Windows:

```
aws docdb modify-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group ^
    --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 ^
    --db-subnet-group-description "Modified subnet group"
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format). Beachten Sie, dass dies die gleiche Subnetzgruppe ist, die im Abschnitt Erstellen einer Amazon DocumentDB Documente-Gruppe erstellt wurde. Die Subnetze in der Subnetzgruppe werden jedoch durch die in der Operation`modify-db-subnet-group` angegebenen ersetzt.

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "DBSubnetGroupDescription": "Modified subnet group",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1d"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-53ab3636"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1b"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-991cb8d0"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1f"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-29ab1025"
            }
        ],
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupName": "sample-subnet-group"
    }
}
```


Löschen einer Amazon DocumentDB Documente-Gruppe

Sie können dasAWS Management Console oder verwenden, um eine Amazon Documente-GruppeAWS CLI zu löschen, um eine Amazon DocumentDB Documente-Gruppe zu löschen. Sie können jedoch nicht die `default`-Subnetzgruppe löschen.

Using the AWS Management Console

Sie können die AWS Management Console verwenden, um eine Subnetzgruppe zu löschen. Die `default`-Subnetzgruppe kann jedoch nicht gelöscht werden.

So löschen Sie eine Subnetzgruppe

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon-DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus. Klicken Sie dann auf die Schaltfläche links neben dem Namen der Subnetzgruppe. Denken Sie daran, dass Sie die `default`-Subnetzgruppe nicht löschen können.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie Actions (Aktionen) und anschließend Delete (Löschen).
4. Im Bestätigungsdialogfeld:
   - Wählen Sie Delete (Löschen), um die Subnetzgruppe zu löschen.
   - Wählen Sie Cancel (Abbrechen), um die Subnetzgruppe zu erhalten.

Using the AWS CLI

Um eine Amazon DocumentDB-Subnetzgruppe mithilfe von zu löschenAWS CLI, verwenden Sie den`delete-db-subnet-group` Vorgang mit dem folgenden Parameter.

Parameter

- `--db-subnet-group-name`—Erforderlich. Der Name der zu löschenden Amazon DocumentDB löschenden Amazon Documentente-Gruppe zu löschen. Denken Sie daran, dass Sie die `default`-Subnetzgruppe nicht löschen können.

Example

Der folgende Code löscht `sample-subnet-group`.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

Für Windows:

```
aws docdb delete-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group
```

Diese Operation erzeugt keine Ausgabe.

Amazon DocumentDB Hochverfügbarkeit und -Replikation

Die Hochverfügbarkeit und Leseskalierung in Amazon DocumentDB (mit MongoDB-Kompatibilität) erreichen Sie durch die Verwendung von Replikat-Instances. Ein einzelner Amazon DocumentDB DocumentDB-Cluster unterstützt eine einzige primäre Instance und bis zu 15 Replikat-Instances. Diese Instances können über mehrere Availability Zones innerhalb der Cluster-Region verteilt werden. Die primäre Instance akzeptiert Lese- und Schreibverkehr, und Replikat-Instances akzeptieren nur Leseanforderungen.

Das Cluster-Volume besteht aus mehreren Kopien der Daten für den Cluster. Die Daten im Cluster-Volume werden jedoch für die primäre Instance und Amazon DocumentDB DocumentDB-Replikate im Cluster als ein einziges logisches Volume dargestellt. Replikat-Instances sind „Eventually Consistent“. Sie geben Abfrageergebnisse mit minimaler Replikatverzögerung zurück – im Normalfall beträgt die Verzögerung weniger als 100 Millisekunden, nachdem die primäre Instance eine Aktualisierung geschrieben hat. Die Replica-Verzögerung variiert in Abhängigkeit vom Veränderungsgrad in der Datenbank. Das heißt, in Zeiten, in denen eine große Anzahl von Schreiboperationen für die Datenbank durchgeführt wird, kann es zu einer Erhöhung der Replikationsverzögerung kommen.

Skalieren von Lesevorgängen

Amazon DocumentDB DocumentDB-Replikate funktionieren für das Skalieren von Lesevorgängen, da sie in Ihrem Cluster-Volume vollständig für Lesevorgänge bereit stehen. Schreibvorgänge werden

von der primären Instance verwaltet. Das Cluster-Volume wird von allen Instances in Ihrem Cluster gemeinsam genutzt. Daher müssen Sie nicht für jedes Amazon DocumentDB DocumentDB-Replikat eine Kopie der Daten replizieren und pflegen.

Hochverfügbarkeit

Wenn Sie ein Amazon DocumentDB DocumentDB-Cluster erstellen, stellt Amazon DocumentDB DocumentDB-Cluster je nach Anzahl an Availability Zones (es müssen mindestens zwei vorhanden sein), Instances in den Availability Zones bereit. Wenn Sie Instances im Cluster erstellen, verteilt Amazon DocumentDB die Instances automatisch in einer Subnetz-Gruppe auf die Availability Zones, um den Cluster auszugleichen. Diese Vorgehensweise verhindert auch, dass sich alle Instances in derselben Availability Zone befinden.

Beispiel

Zur Veranschaulichung nehmen wir ein Beispiel, in dem ein Cluster erstellt wird, das eine Subnetzgruppe mit drei Availability Zones hat Availability Zones: AZ1,AZ2, undAZ3aus.

Wenn die erste Instance im Cluster erstellt wird, ist diese die primäre Instance und befindet sich in einer der Availability Zones. In diesem Beispiel handelt es sich um AZ1. Die zweite erstellte Instance ist eine Replikat-Instance und befindet sich in einer der beiden anderen Availability Zones, z.B. AZ2. Die dritte Instance, die erstellt wird, ist eine Replikat-Instance und befindet sich in der verbleibenden Availability ZoneAZ3. Wenn Sie mehrere Instances erstellen, werden diese über die Availability Zones verteilt, sodass der Cluster ausbalanciert ist.

Tritt in der primären Instance (AZ1) ein Fehler auf, wird ein Failover ausgelöst und eine der bestehenden Replikat-Instances wird zur primären Instance heraufgestuft. Wenn die alten primäre Instance wiederhergestellt wurde, wird sie zu einem Replikat in der gleichen Availability Zone, in der sie bereitgestellt wurde (AZ1). Wenn Sie ein Cluster mit drei Instances bereitstellen, behält Amazon DocumentDB diesen Cluster mit drei Instances weiter bei. Amazon DocumentDB übernimmt automatisch Erkennung, Failover und Wiederherstellung von Instance-Fehlern, ohne dass ein manueller Eingriff erforderlich ist.

Wenn Amazon DocumentDB ein Failover durchführt und eine Instance wiederherstellt, bleibt die wiederhergestellte Instance in der Availability Zone, in der sie ursprünglich bereitgestellt wurde. Die Rolle der Instance ändert sich möglicherweise von der primären Instance zum Replikat. Dadurch wird ein Szenario verhindert, in dem eine Reihe von Failovers in allen Instances auftreten könnten, die sich in derselben Availability Zone befinden.

Sie können Amazon DocumentDB DocumentDB-Replikate als Failover-Ziele angeben. Das heißt, wenn die primäre Instance ausfällt, wird das angegebene Amazon DocumentDB DocumentDB-Replikat oder -Replikat von einer Stufe auf die primäre Instance heraufgestuft. Es gibt dann eine kurze Unterbrechung, während der Lese- und Schreibanfragen an die primäre Instance mit einer Ausnahme fehlschlagen. Wenn Ihr Amazon DocumentDB DocumentDB-Cluster keine Amazon DocumentDB DocumentDB-Replikate enthält, wird sie bei Ausfall der primären Instance neu erstellt. Die Hochstufung eines Amazon DocumentDB DocumentDB-Replikats ist viel schneller als die Neuerstellung der primären Instance

Für Szenarios mit hoher Verfügbarkeit empfehlen wir Ihnen, mindestens ein Amazon DocumentDB DocumentDB-Replikat zu erstellen. Diese Replikate sollten von der gleichen Instance-Klasse wie die primäre Instance und in verschiedenen Availability Zones für Ihren Amazon DocumentDB DocumentDB-Cluster sein.

Weitere Informationen finden Sie unter:

- Grundlegendes zur Fehlertoleranz des Amazon DocumentDB-Clusters
- Amazon DocumentDB DocumentDB-Failover
  - Steuern des Failover-Ziels

Hochverfügbarkeit mit globalen Clustern

Für hohe Verfügbarkeit über mehrereAWS-Regionenkönnen Sie einrichtenGlobale Amazon DocumentDB DocumentDB-Clusteraus. Jeder globale Cluster erstreckt sich über mehrere -Regionen und ermöglicht globales Lesen mit geringer Latenz sowie eine Notfallwiederherstellung nach AusfällenAWS-Regionaus. Amazon DocumentDB übernimmt automatisch die Replikation aller Daten und Aktualisierungen aus der primären -Region in jede der sekundären Regionen.

Hinzufügen von -Replicas

Die erste dem Cluster hinzugefügte Instance ist die primäre Instance. Jede Instance, die nach der ersten Instance hinzugefügt wird, ist eine Replikat-Instance. Ein Cluster kann zusätzlich zur primären bis zu 15 -Replikat-Instances haben.

Wenn Sie einen Cluster mit der AWS Management Console erstellen, wird gleichzeitig automatisch eine primäre Instance angelegt. Um ein Replikat gleichzeitig mit der Erstellung des Clusters und der primären Instance zu erstellen, wählen Sie Create replica in different zone (Replikat in unterschiedlicher Zone erstellen) aus. Weitere Informationen finden Sie in Schritt 4.d unter Erstellen

eines Amazon DocumentDB-Clusters. Wenn Sie weitere Replikate zu einem Amazon DocumentDB DocumentDB-Cluster hinzufügen möchten, finden Sie weitere Informationen unterHinzufügen einer Amazon DocumentDB-Instance zu einem Clusteraus.

Wenn Sie die AWS CLI zum Erstellen Ihres Clusters verwenden, müssen Sie Ihre primäre Instance und die Replikat-Instance explizit erstellen. Weitere Informationen finden Sie im Abschnitt "Verwendung der AWS CLI" unter der folgenden Themen:

- Erstellen eines Amazon DocumentDB-Clusters
- Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster

Amazon DocumentDB DocumentDB-Failover

In bestimmten Fällen, wie z. B. bei bestimmten Arten von planmäßiger Wartung oder im unwahrscheinlichen Fall eines Ausfalls eines primären Knotens oder einer Availability Zone, erkennt Amazon DocumentDB DocumentDBs (mit MongoDB-Kompatibilität) den Ausfall und ersetzt den primären Knoten. Während eines Failovers wird die Ausfallzeit für Schreibvorgänge minimiert. Das liegt daran, dass die Rolle des primären Knotens auf eine der Read Replicas übergeht, statt dass ein neuer primärer Knoten erstellt und bereitgestellt werden muss. Durch Ausfallerkennung und Replikatheraufstufung wird sichergestellt, dass Sie weiter in den neuen primären Knoten schreiben können, sobald die Heraufstufung abgeschlossen wurde.

Damit das Failover funktioniert, muss Ihr Cluster mindestens zwei Instanzen haben - eine primäre und mindestens eine Replikatinstanz.

Steuern des Failover-Ziels

Amazon DocumentDB DocumentDB-Stufen stellt Ihnen Failover-Stufen zur Verfügung, um zu steuern, welche Replikat-Instance bei einem Failover auf primär umgestellt wird.

Failover-Stufen

Jede Replikat-Instance ist eine Failover-Stufe (0-15) zugeordnet. Wenn ein Failover aufgrund von Wartung oder einem unwahrscheinlichen Hardwareausfall auftritt, geht die primäre Instance auf ein Replikat mit der höchsten Priorität (die niedrigste nummerierte Stufe) über. Wenn mehrere Replikate die gleiche Prioritätsstufe haben, geht die primäre Instance auf das Replikat dieser Stufe über, die der Größe der vorherigen primären Instance am nächsten kommt.

Indem Sie die Failover-Stufe für eine Gruppe ausgewählter Replikate auf `0` (höchste Priorität) setzen, können Sie sicherstellen, dass ein Failover auf eines der Replikate in dieser Gruppe wechselt. Sie können effektiv verhindern, dass bestimmte Replikate im Falle eines Failover zur primären Instance hochgestuft werden, indem Sie diesen Replikaten eine niedrige Stufe (hohe Anzahl) zuweisen. Dies ist nützlich, wenn bestimmte Replikate von einer Anwendung stark genutzt werden und ein Failover auf eine von ihnen eine kritische Anwendung negativ beeinflussen würde.

Sie können die Failover-Stufe einer Instance beim Erstellen oder später festlegen. Das Festlegen einer Instance-Failover-Stufe durch Ändern der Instance löst keinen Failover aus. Weitere Informationen finden Sie in den folgenden Themen:

- Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster
- Ändern einer Amazon DocumentDB-Instance

Wenn Sie einen Failover manuell einleiten, haben Sie zwei Möglichkeiten zur Steuerung, welche Replikat-Instance auf primär umgestellt wird: die Failover-Stufen wie zuvor beschrieben und den Parameter `--target-db-instance-identifier`.

`--target-db-instance-identifier`

Zum Testen können Sie mit der Operation `failover-db-cluster` ein Failover-Ereignis erzwingen. Mit dem Parameter `--target-db-instance-identifier` können Sie festlegen, welches Replikat auf primär umgestellt werden soll. Die Verwendung des Parameters `--target-db-instance-identifier` ersetzt die Failover-Prioritätsstufe. Wenn Sie den Parameter `--target-db-instance-identifier` nicht angeben, entspricht die Primär-Failover-Funktion der Failover-Prioritätsstufe.

Was passiert während eines Failovers?

Der Failover wird automatisch von Amazon DocumentDB DocumentDBs durchgeführt, sodass Ihre Anwendungen den Datenbankbetrieb so schnell wie möglich und ohne Verwaltungseingriff wieder aufnehmen können.

- Wenn Sie beim Ausfall eine Amazon DocumentDB Replikatinstanz in derselben oder einer anderen Availability Zone haben: Amazon DocumentDB DocumentDB-Limits wechselt den anerkannten Namensdatensatz (CNAME) für Ihre Instance, sodass auf das fehlerfreie Replikat verweist, das dadurch zur neuen primären Instance hochgestuft wird. Das Failover wird in der Regel innerhalb von 30 Sekunden vom Anfang bis zum Ende abgeschlossen.

- Wenn Sie keine Amazon DocumentDB DocumentDB-Replikat-Instance haben (z. B. einen einzelnen Instance-Cluster): Amazon DocumentDB DocumentDB-Stufe wird versuchen, eine neue Instance in derselben Availability Zone wie die ursprüngliche Instance zu erstellen. Dieser Austausch der ursprünglichen Instance wird nach bestem Bemühen durchgeführt, ist aber nicht immer erfolgreich, z. B. wenn ein Problem vorliegt, das sich allgemein auf die Availability Zone auswirkt.

Bei Verbindungsunterbrechung muss Ihre Anwendung versuchen, die Verbindung zur Datenbank wiederherzustellen.

Testen eines Failovers

Ein Failover für einen Cluster stuft eine der Amazon DocumentDB DocumentDB-Replikate (Read-Only-Instances) im Cluster zu einer primären Instance (den Cluster-Writer) fest.

führt automatisch einen Failover auf ein Amazon DocumentDB DocumentDB-Replikat aus (falls vorhanden), sobald die primäre Instance ausfällt. Sie können ein Failover erzwingen, wenn Sie einen Ausfall einer primären Instance zum Testen simulieren möchten. Jede Instance in einem Cluster hat eine eigene Endpunkt-Adresse. Aus diesem Grund müssen Sie alle bestehenden Verbindungen, die diese Endpunktadressen verwenden, bereinigen und wiederherstellen, wenn der Failover abgeschlossen ist.

Um einen Failover zu erzwingen, verwenden Sie die Operation `failover-db-cluster` mit diesen Parametern.

- `--db-cluster-identifier`—Erforderlich. Der Name des Clusters, der einen Failover durchführen soll.
- `--target-db-instance-identifier`—Optional. Der Name der Instance, die zur primären Instance befördert werden soll.

Example

Die folgende Operation erzwingt einen Failover des Clusters `sample-cluster`. Es wird nicht angegeben, welche Instance die neue primäre Instance bilden soll, daher wählt Amazon DocumentDB DocumentDB-Limits die Instance entsprechend der Failover-Tier-Stufe aus.

Für Linux, macOS oder Unix:

```
aws docdb failover-db-cluster \
```

```
    --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb failover-db-cluster ^
    --db-cluster-identifier sample-cluster
```

Die folgende Operation erzwingt einen Failover des Clusters `sample-cluster` und legt fest, dass `sample-cluster-instance` in die primäre Rolle befördert werden soll. (Beachten Sie `"IsClusterWriter": true` in der Ausgabe.)

Für Linux, macOS oder Unix:

```
aws docdb failover-db-cluster \
    --db-cluster-identifier sample-cluster \
    --target-db-instance-identifier sample-cluster-instance
```

Für Windows:

```
aws docdb failover-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --target-db-instance-identifier sample-cluster-instance
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBCluster": {
        "HostedZoneId": "Z2SUY0A1719RZT",
        "Port": 27017,
        "EngineVersion": "3.6.0",
        "PreferredMaintenanceWindow": "thu:04:05-thu:04:35",
        "BackupRetentionPeriod": 1,
        "ClusterCreateTime": "2018-06-28T18:53:29.455Z",
        "AssociatedRoles": [],
        "DBSubnetGroup": "default",
        "MasterUsername": "master-user",
        "Engine": "docdb",
        "ReadReplicaIdentifiers": [],
        "EarliestRestorableTime": "2018-08-21T00:04:10.546Z",
        "DBClusterIdentifier": "sample-cluster",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "DBClusterMembers": [
```

```
            {
                "DBInstanceIdentifier": "sample-cluster-instance",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": true
            },
            {
                "DBInstanceIdentifier": "sample-cluster-instance-00",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": false
            },
            {
                "DBInstanceIdentifier": "sample-cluster-instance-01",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": false
            }
        ],
        "AvailabilityZones": [
            "us-east-1b",
            "us-east-1c",
            "us-east-1a"
        ],
        "DBClusterParameterGroup": "default.docdb3.6",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "IAMDatabaseAuthenticationEnabled": false,
        "AllocatedStorage": 1,
        "LatestRestorableTime": "2018-08-22T21:57:33.904Z",
        "PreferredBackupWindow": "00:00-00:30",
        "StorageEncrypted": false,
        "MultiAZ": true,
        "Status": "available",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-12345678"
            }
        ],
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    }
}
```

Replikationsverzögerung

Die Replikationsverzögerung beträgt normalerweise 50 ms oder weniger. Die häufigsten Gründe für eine erhöhte Replikat-Verzögerung sind:

- Eine hohe Schreibrate für die Primärwaffe, die dazu führt, dass die Read-Replikate hinter die primäre zurückbleiben.
- Streit bei den Lese-Replikaten zwischen lang laufenden Abfragen (z. B. große sequenzielle Scans, Aggregationsabfragen) und eingehender Schreibreplikation.
- Sehr große Anzahl gleichzeitiger Abfragen zu den Read-Replikaten.

Um die Replikationsverzögerung zu minimieren, versuchen Sie diese Techniken zur Fehlerbehebung:

- Wenn Sie eine hohe Schreibrate oder eine hohe CPU-Auslastung haben, empfehlen wir Ihnen, die Instanzen in Ihrem Cluster zu skalieren.
- Wenn es lang andauernde Abfragen zu Ihren Lese-Replikaten und sehr häufige Aktualisierungen der abgefragten Dokumente gibt, sollten Sie in Betracht ziehen, Ihre lang andauernden Abfragen zu ändern oder sie mit dem Primär-/Schreibreplikat auszuführen, um Konflikte bei den Lese-Replikaten zu vermeiden.
- Wenn nur bei den Lese-Replikaten eine sehr große Anzahl gleichzeitiger Abfragen oder eine hohe CPU-Auslastung vorhanden ist, besteht eine andere Möglichkeit darin, die Anzahl der Lese-Replikate zu skalieren, um die Arbeitslast zu verteilen.
- Da die Replikationsverzögerung auf einen hohen Schreibdurchsatz und lang laufende Abfragen zurückzuführen ist, empfehlen wir, die Replikationsverzögerung zu beheben, indem Sie die DbClusterReplicalAgMaximum CW-Metrik in Kombination mit dem langsamen Abfragelogger und`WriteThroughput`/`WriteIOPS`-Metriken.

Im Allgemeinen empfehlen wir, dass alle Ihre Replikate vom gleichen Instanztyp sind, damit ein Cluster-Failover keine Leistungsverschlechterung verursacht.

Wenn Sie sich zwischen Skalierung und Skalierung entscheiden (z. B. sechs kleinere Instanzen im Vergleich zu drei größeren Instanzen), empfehlen wir im Allgemeinen, zuerst (größere Instanzen) zu skalieren, bevor Sie einen größeren Puffer-Cache pro DB-Instance erhalten.

Proaktiv sollten Sie einen Alarm für die Replikationsverzögerung einstellen und seinen Schwellenwert auf einen Wert festlegen, von dem Sie glauben, dass er die Obergrenze dafür ist, wie weit Ihre

Daten auf Replikatinstanzen hinter (oder „veraltet“) liegen können, bevor sie die Funktionalität Ihrer Anwendung beeinträchtigen. Im Allgemeinen empfehlen wir, dass der Schwellenwert für die Replikationsverzögerung für mehrere Datenpunkte vor Alarmierung aufgrund vorübergehender Workloads überschritten wird.

> **Note**
>
> Darüber hinaus empfehlen wir Ihnen, einen weiteren Alarm für Replikationsverzögerungen einzustellen, die 10 Sekunden überschreiten. Wenn Sie diesen Schwellenwert für mehrere Datenpunkte überschreiten, empfehlen wir Ihnen, Ihre Instanzen zu vergrößern oder Ihren Schreibdurchsatz für die primäre Instance zu reduzieren.

Verwalten von Amazon DocumentDB-Indizes

Amazon DocumentDB-Indexerstellung

Für die Erstellung von Indizes in Amazon DocumentDB müssen eine Reihe von Entscheidungen getroffen werden:

- Wie schnell muss es abgeschlossen werden?
- Kann auf die Sammlung nicht zugegriffen werden, während der Build stattfindet?
- Wie viel Rechenleistung einer Instance kann dem Build zugewiesen werden?
- Welche Art von Index sollte erstellt werden?

Dieser Abschnitt hilft Ihnen bei der Beantwortung dieser Fragen und enthält die Befehle und Überwachungsbeispiele zum Erstellen eines Amazon DocumentDB-Index in Ihrer Instance-basierten Clustersammlung.

Richtlinien

Die folgenden Richtlinien enthalten grundlegende Limits und Konfigurationsnachteile beim Erstellen neuer Indizes:

- Amazon DocumentDB-Versionsunterstützung – Während die Indizierung einzelner Worker in allen Amazon DocumentDB-Versionen unterstützt wird, wird die Indizierung mehrerer Worker nur in den Amazon DocumentDB-Versionen 4.0 und 5.0 unterstützt.

- Leistungsnachteil – Eine Erhöhung der Anzahl der Worker im Indexerstellungsprozess erhöht die CPU-Auslastung und den Lese-IO auf der primären Instance Ihrer Amazon DocumentDB-Datenbank. Die Ressourcen, die zum Erstellen eines neuen Indexes benötigt werden, sind für Ihren laufenden Workload nicht verfügbar.
- Elastic Cluster – Parallele Indizierung wird auf elastischen Amazon DocumentDB-Clustern nicht unterstützt.
- Maximale Worker – Die maximale Anzahl von Workern, die Sie konfigurieren können, hängt von der Größe Ihrer primären Instance in Ihrem Datenbank-Cluster ab. Dies ist die Hälfte der Gesamtzahl der vCPUs auf der primären Instance Ihres Datenbank-Clusters. Sie können beispielsweise maximal 32 Worker auf einer db.r6g.16xlarge-Instance mit 64 vCPUs ausführen.

  > **Note**
  >
  > Parallele Worker werden auf 2xlarge-Instance-Klassen und niedriger nicht unterstützt.

- Minimale Anzahl von Workern – Die Mindestanzahl von Workern, die Sie konfigurieren können, ist eins. Die Standardeinstellung für die Indexerstellung auf Instance-basierten Clustern sind zwei Worker. Sie können jedoch die Anzahl der Worker auf einen reduzieren, indem Sie die Option „Worker-Threads“ verwenden. Dadurch wird der Prozess mit einem einzigen Worker ausgeführt.
- Indexkomprimierung – Amazon DocumentDB unterstützt keine Indexkomprimierung. Die Datengrößen für Indizes können größer sein als bei Verwendung anderer Optionen.
- Indizierung mehrerer Sammlungen – Die vCPUs auf der primären Instance Ihres Datenbank-Clusters können für konfigurierte Worker verwendet werden, die die Indexerstellung für mehrere Sammlungen durchführen.
- Indextypen – In diesem Blogbeitrag finden Sie eine vollständige Erklärung der unterstützten Indextypen in Amazon DocumentDB.

Erste Schritte

Um die Indexerstellung für eine Sammlung zu starten, verwenden Sie die `-db.collection.createIndex()`Methode aus der mongo-Shell, um Indizes zu erstellen. Standardmäßig führt der Befehl zwei parallele Worker aus, die die Geschwindigkeit des Indexerstellungsprozesses um das Zweifache erhöhen.

Der folgende Befehlsprozess zeigt beispielsweise, wie Sie einen Index für das Feld „user_name“ in einem Dokument erstellen und die Indizierungsprozessgeschwindigkeit auf vier Worker erhöhen:

1. Erstellen Sie Indizes mit zwei parallelen Workern im Cluster:

```
db.test.createIndex({user_name:1})
```

2. Um die Geschwindigkeit des Indexerstellungsprozesses zu optimieren, können Sie die Anzahl der Worker mithilfe der Option „Worker-Threads“ (`workers:number`) in der `db.collection.createIndex()` Methode angeben.

   Erhöhen Sie die Geschwindigkeit des Prozesses auf vier parallele Worker:

```
db.test.createIndex({user_name:1}, {workers:4})
```

   > **Note**
   >
   > Je höher die Anzahl der Worker ist, desto schneller wird die Indexerstellung ausgeführt. Je höher die Anzahl der Worker ist, desto höher steigt die Last der vCPUs und Lese-IO Ihrer primären Instance. Stellen Sie sicher, dass Ihr Cluster ausreichend bereitgestellt ist, um den erhöhten Aufwand zu bewältigen, ohne andere Workloads zu beeinträchtigen.

Status des Indizierungsfortschritts

Der Indexerstellungsprozess funktioniert durch Initialisieren, Scannen von Sammlungen, Sortierschlüsseln und schließlich das Einfügen von Schlüsseln mithilfe eines Index Builders. Der Prozess hat bis zu sechs Phasen, wenn Sie ihn im Vordergrund ausführen, und bis zu neun Phasen, wenn Sie ihn im Hintergrund ausführen. Sie können Statusmetriken wie den prozentualen Abschluss, die Gesamtzahl der gescannten Speicherblöcke, sortierte Schlüssel und eingefügte Schlüssel auf Stufenbasis anzeigen.

Überwachen Sie den Fortschritt des Indizierungsprozesses, indem Sie den `db.currentOp()` Befehl in der mongo-Shell verwenden. Ein 100%iger Abschluss der letzten Phase zeigt, dass alle Indizes erfolgreich erstellt wurden:

```
db.currentOp({"command.createIndexes": { $exists : true } })
```

Index-Build-Typen

Die vier Arten von Index-Builds sind:

- Vordergrund – Der Vordergrundindex-Build blockiert alle anderen Datenbankoperationen, bis der Index erstellt ist. Der AmazonAmazon DocumentDB-Foreground-Build besteht aus fünf Phasen.
- Vordergrund (eindeutig) – Einzeldokument- (eindeutige) Vordergrundindex-Builds blockieren andere Datenbankoperationen wie reguläre Vordergrund-Builds. Im Gegensatz zum grundlegenden Vordergrund-Build verwendet der eindeutige Build eine zusätzliche Stufe (Sortierschlüssel 2), um nach doppelten Schlüsseln zu suchen. Der (eindeutige) Vordergrund-Build besteht aus sechs Phasen.
- Hintergrund – Der Hintergrundindex-Build ermöglicht die Ausführung anderer Datenbankoperationen im Vordergrund, während der Index erstellt wird. Der Amazon DocumentDB-Hintergrundaufbau besteht aus acht Phasen.
- Hintergrund (eindeutig) – Einzelne Dokument-Hintergrundindex-Builds (eindeutig) ermöglichen die Ausführung anderer Datenbankoperationen im Vordergrund, während der Index erstellt wird. Im Gegensatz zum grundlegenden Hintergrund-Build verwendet der eindeutige Build eine zusätzliche Stufe (Sortierschlüssel 2), um nach doppelten Schlüsseln zu suchen. Der (eindeutige) Hintergrundaufbau besteht aus neun Phasen.

Index-Build-Phasen

Stufe	Vordergrund	Vordergrund (eindeutig)	Hintergrund	Hintergrund (eindeutig)
Initialisieren	1	1	1	1
Building Index: initialisieren	2	2	2	2
Bauindex: collectio scannen	3	3	3	3
Bauindex: Sortierschlüssel 1	4	4	4	4
Bauindex: Sortierschlüssel 2		5		5

Stufe	Vordergrund	Vordergrund (eindeutig)	Hintergrund	Hintergrund (eindeutig)
Bauindex: Einfügen von Schlüsseln	5	6	5	6
Validieren: Scan-Index			6	7
Validieren: Sortieren von Tupels			7	8
Validieren: Scannen der Sammlung			8	9

- initialisieren – createIndex bereitet den Index Builder vor. Diese Phase sollte sehr kurz sein.
- Building Index: initialisieren – Der Index Builder bereitet die Erstellung des Index vor. Diese Phase sollte sehr kurz sein.
- building index: scan collection – Der Index Builder führt einen Sammlungsscan durch, um Indexschlüssel zu erfassen. Die Maßeinheit ist „Blöcke“.

> **Note**
>
> Wenn mehr als ein Worker für den Indexaufbau konfiguriert ist, wird er in dieser Phase angezeigt. Die Phase „Scannen der Sammlung“ ist die einzige Phase, in der während des Indexerstellungsprozesses mehrere Worker verwendet werden. In allen anderen Phasen wird ein einzelner Worker angezeigt.

- building index: Sortierschlüssel 1 – Der Index Builder sortiert die erfassten Indexschlüssel. Die Maßeinheit ist „Schlüssel“.
- Bauindex: Sortierschlüssel 2 – Der Index-Builder sortiert die erfassten Indexschlüssel, die toten Tupeln entsprechen. Diese Phase ist nur für die Erstellung eines eindeutigen Indexes vorhanden. Die Maßeinheit ist „Schlüssel“.

- building index: inserting keys – Der Index Builder fügt Indexschlüssel in den neuen Index ein. Die Maßeinheit ist „Schlüssel“.
- validieren: scan index – createIndex scannt den Index, um Schlüssel zu finden, die validiert werden müssen. Die Maßeinheit ist „Blöcke“.
- Validierung: Sortiert Tupel – createIndex sortiert die Ausgabe der Indexscanphase.
- validieren: Sammlung scannen – createIndex scannt die Sammlung, um die Indexschlüssel zu validieren, die in den beiden vorherigen Phasen gefunden wurden. Die Maßeinheit ist „Blöcke“.

Beispiel für eine Index-Build-Ausgabe

Im folgenden Ausgabebeispiel (Erstellung des Vordergrundindex) wird der Status der Indexerstellung angezeigt. Das Feld „msg“ fasst den Build-Fortschritt zusammen, indem die Stufe und der Abschlussprozentsatz des Builds angegeben werden. Das Feld „Worker“ gibt die Anzahl der Worker an, die in dieser Phase des Indexaufbaus verwendet wurden. Das Feld „Fortschritt“ zeigt die tatsächlichen Zahlen an, die zur Berechnung des Prozentsatzes des Abschlusses verwendet werden.

> **Note**
>
> Die Felder „currentIndexBuildName“, „msg“ und „progress“ werden in Amazon DocumentDB Version 4.0 nicht unterstützt.

```
{
    "inprog" : [{
    …
        "command": {
            "createIndexes": "test",
            "indexes": [{
                "v": 2,
                "key": {
                    "user_name": 1
                },
                "name": "user_name_1"
            }],
            "lsid": {
                "id": UUID(“094d0fba-8f41-4373-82c3-7c4c7b5ff13b”)
            },
            "$db": "test"
        },
```

```
        "currentIndexBuildName": user_name_1,
        "msg": "Index Build: building index number_1, stage 6/6 building index:
 656860/1003520 (keys) 65%",
        "workers": 1,
        "progress": {
            "done": 656861,
            "total": 1003520
        },
    …
    ],

    "ok" : 1
}
```

Verwaltung der Dokumentenkomprimierung auf Sammlungsebene

Die Amazon DocumentDB-Dokumentkomprimierung auf Sammlungsebene ermöglicht es Ihnen, die Speicher- und I/O-Kosten zu senken, indem Sie die Dokumente in Ihren Sammlungen komprimieren. Sie können die Dokumentenkomprimierung auf Sammlungsebene aktivieren und die Kompressionsmetriken nach Bedarf einsehen, indem Sie die Speichergewinne anhand von Kompressionsmetriken wie der Speichergröße komprimierter Dokumente und dem Komprimierungsstatus messen. Amazon DocumentDB verwendet den LZ4-Komprimierungsalgorithmus, um Dokumente zu komprimieren.

Richtlinien

Die folgenden Richtlinien gelten für die Komprimierung von Dokumenten auf Sammlungsebene:

- Die Dokumentenkomprimierung ist standardmäßig deaktiviert
- Die Dokumentenkomprimierung kann nicht auf bestehende Sammlungen angewendet werden.
- Die Dokumentenkomprimierung wird nur in Amazon DocumentDB Version 5.0 und höher unterstützt.
- Amazon DocumentDB komprimiert nur Dokumente mit einer Größe von 2 KB und mehr.

Dokumentenkomprimierung aktivieren

Aktivieren Sie die Dokumentenkomprimierung beim Erstellen einer Sammlung in Amazon DocumentDB mithilfe der folgenden `db.createCollection()` Methode:

```
db.createCollection( sample_collection,{
    storageEngine : {
        documentDB: {
            compression:{
                enable: <true | false>
            }
        }
    }
})
```

Überwachung der Dokumentenkomprimierung

Sie können überprüfen, ob eine Sammlung komprimiert ist, und ihr Kompressionsverhältnis wie folgt berechnen.

Sehen Sie sich die Kompressionsstatistiken an, indem Sie den `db.collection.stats()` Befehl `db.printCollectionStats()` or in der Mongo-Shell ausführen. Die Ausgabe zeigt Ihnen die Originalgröße und die komprimierte Größe, die Sie vergleichen können, um die Speichergewinne durch die Dokumentenkomprimierung zu analysieren. In diesem Beispiel werden Statistiken für eine Sammlung mit dem Namen „sample_collection“ angezeigt:

```
db.sample_collection.stats(1024*1024)

{
    "ns" : "test.sample_collection",
    "count" : 1000000,
    "size" : 3906.3,
    "avgObjSize" : 4096,
    "storageSize" : 1953.1,
    compression:{
        "enabled" : true,
        "threshold" : 2032
    }
    ...
}
```

- Größe — Die Originalgröße der Dokumentensammlung.
- avgObjSize- Die durchschnittliche Dokumentengröße vor der Komprimierung, auf die erste Dezimalzahl gerundet. Die Maßeinheit ist Byte.

- storageSize — Die Speichergröße der Sammlung nach der Komprimierung. Die Maßeinheit ist Byte.
- aktiviert — Zeigt an, ob die Komprimierung aktiviert oder deaktiviert ist.

Um das tatsächliche Komprimierungsverhältnis zu berechnen, dividieren Sie die Sammlungsgröße durch die Speichergröße (Size/StorageSize). Für das obige Beispiel lautet die Berechnung 3906,3/1953,1, was einem Kompressionsverhältnis von 2:1 entspricht.

Verwaltung vorhandener Sammlungen

Sie können zwar eine bestehende Sammlung nicht komprimieren, aber Sie können unkomprimierte oder komprimierte Dokumente konvertieren. Um vorhandene unkomprimierte Dokumente im komprimierten Format zu speichern, kopieren Sie das Dokument in eine komprimierungsfähige Sammlung. Um komprimierte Dokumente in ein unkomprimiertes Format zu konvertieren, kopieren Sie die Dokumente in eine Sammlung, für die die Komprimierung deaktiviert ist.

Verwalten von Amazon DocumentDB DocumentDB-Ereignissen

Amazon DocumentDB (mit MongoDB-Kompatibilität) speichert den Verlauf von Ereignissen, die mit Ihren Clustern, Instances, Snapshots, Sicherheitsgruppen und ClusterParametergruppen zusammenhängen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp des Ereignisses und eine dem Ereignis zugehörige Benachrichtigung.

> **Important**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS und Amazon Neptune gemeinsam genutzt wird. Regionsgrenzen, d. h. Grenzwerte, die auf Regionsebene geregelt werden, werden von Amazon DocumentDB, Amazon RDS und Amazon Neptune gemeinsam genutzt. Weitere Informationen finden Sie unter Regionale Kontingente.

Themen

- Amazon DocumentDB
- Amazon DocumentDB

Amazon DocumentDB

Jeder Amazon DocumentDB DocumentDB-Ressourcentyp hat bestimmte Ereignistypen, die ihm zugeordnet werden können. Sie können denAWS CLI`describe-event-categories` Vorgang verwenden, um die Zuordnung zwischen Ereignistypen und Amazon DocumentDB DocumentDB-Ressourcentypen anzuzeigen.

Parameter

- **`--source-type`**—Fakultativ. Verwenden Sie den `--source-type`-Parameter, um die Ereigniskategorien für einen bestimmten Quelltyp anzuzeigen. Die folgenden Werte sind zulässig:
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`
- **`--filters`**—Fakultativ. Verwenden Sie den Filter, um die Veranstaltungskategorien nur für Amazon DocumentDB anzuzeigen`--filter Name=engine,Values=docdb`.

Example

Der folgende Code listet die Ereigniskategorien im Zusammenhang mit Clustern auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-event-categories \
    --filter Name=engine,Values=docdb \
    --source-type db-cluster
```

Für Windows:

```
aws docdb describe-event-categories ^
    --filter Name=engine,Values=docdb ^
    --source-type db-cluster
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
```

```
    "EventCategoriesMapList": [
        {
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
            ],
            "SourceType": "db-cluster"
        }
    ]
}
```

Der folgende Code listet die Ereigniskategorien auf, die jedem Amazon DocumentDB DocumentDB-Quelltyp zugeordnet sind.

```
aws docdb describe-event-categories
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-instance",
            "EventCategories": [
                "notification",
                "failure",
                "creation",
                "maintenance",
                "deletion",
                "recovery",
                "restoration",
                "configuration change",
                "read replica",
                "backtrack",
                "low storage",
                "backup",
                "availability",
                "failover"
            ]
        },
        {
            "SourceType": "db-security-group",
```

```
            "EventCategories": [
                "configuration change",
                "failure"
            ]
        },
        {
            "SourceType": "db-parameter-group",
            "EventCategories": [
                "configuration change"
            ]
        },
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
            ]
        },
        {
            "SourceType": "db-cluster-snapshot",
            "EventCategories": [
                "backup"
            ]
        }
    ]
}
```

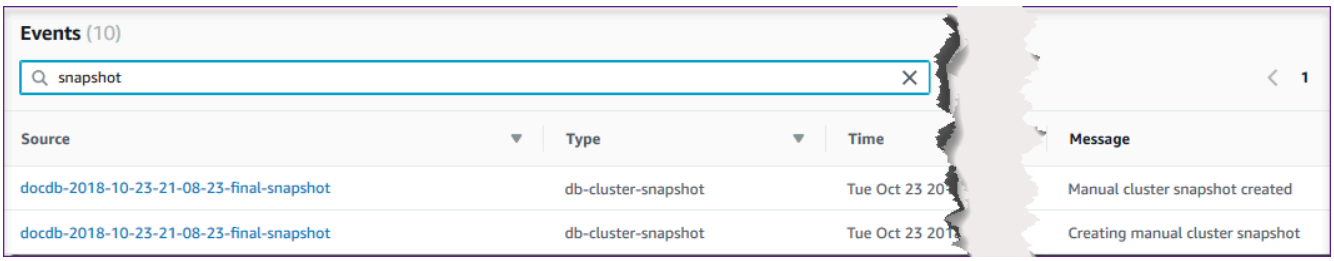
Amazon DocumentDB

Ereignisse aus Ihren Amazon-DocumentDB-Ressourcen können Ereignisse aus Ihren letzten 24 Stunden Sie können Ereignisse aus Ihren Amazon-DocumentDB-Ressourcen auch abrufen, indem Sie denAWS CLI -Befehl describe-events oder die DescribeEventsAmazon-DocumentDB-API-Operation Wenn SieAWS CLI oder Amazon-DocumentDB-API verwenden, Ereignisse, können Ereignisse aus den letzten 14 Tagen abrufen.

Using the AWS Management Console

So können Sie alle Amazon-DocumentDB-Instance-Ereignisse der letzten 24 Stunden

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon-DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich die Option Events. Die verfügbaren Ereignisse erscheinen in einer Liste.
3. Verwenden Sie die Liste Filter, um die Ereignisse nach Typ zu filtern. Geben Sie einen Begriff in das Textfeld ein, um Ihre Ergebnisse weiter zu filtern. Der folgende Screenshot zeigt beispielsweise, wie alle Amazon-DocumentDB-Ereignisse nach Snapshot-Ereignissen.



Using the AWS CLI

So können Sie alle Amazon-DocumentDB-Instance-Ereignisse der letzten 7 Tage

Sie können alle Amazon-DocumentDB-Instance-EreignisseAWS CLI der`--duration` letzten 7 Tage`10080`

```
aws docdb describe-events --duration 10080
```

Filterung nach Amazon DocumentDB DocumentDB-Ereignissen

Verwenden Sie den`describe-events` Vorgang mit den folgenden Parametern, um bestimmte Amazon DocumentDB DocumentDB-Ereignisse zu sehen.

Parameter

- **`--filter`**— Erforderlich, um die zurückgegebenen Werte auf Amazon DocumentDB DocumentDB-Ereignisse zu beschränken. Wird verwendet**`Name=engine,Values=docdb`**, um alle Ereignisse nur für Amazon DocumentDB zu filtern.

- **`--source-identifier`**—Fakultativ. ID der Ereignisquelle, für die Ereignisse zurückgegeben werden. Wenn diese Option weggelassen wird, werden Ereignisse aus allen Quellen in die Ergebnisse einbezogen.
- **`--source-type`**— Optional, sofern nicht`--source-identifier` anders angegeben, dann erforderlich. Wenn `--source-identifier` angegeben wird, muss `--source-type` mit dem Typ der `--source-identifier` übereinstimmen. Die folgenden Werte sind zulässig:
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`

Im folgenden Beispiel werden alle Amazon-DocumentDB-Ereignisse

```
aws docdb describe-events --filters Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "Events": [
        {
            "SourceArn": "arn:aws:rds:us-east-1:123SAMPLE012:db:sample-cluster-
instance3",
            "Message": "instance created",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:17:40.023Z",
            "SourceIdentifier": "sample-cluster-instance3",
            "EventCategories": [
                "creation"
            ]
        },
        {
            "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
            "Message": "instance shutdown",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:25:01.245Z",
            "SourceIdentifier": "docdb-2018-12-11-21-08-23",
            "EventCategories": [
```

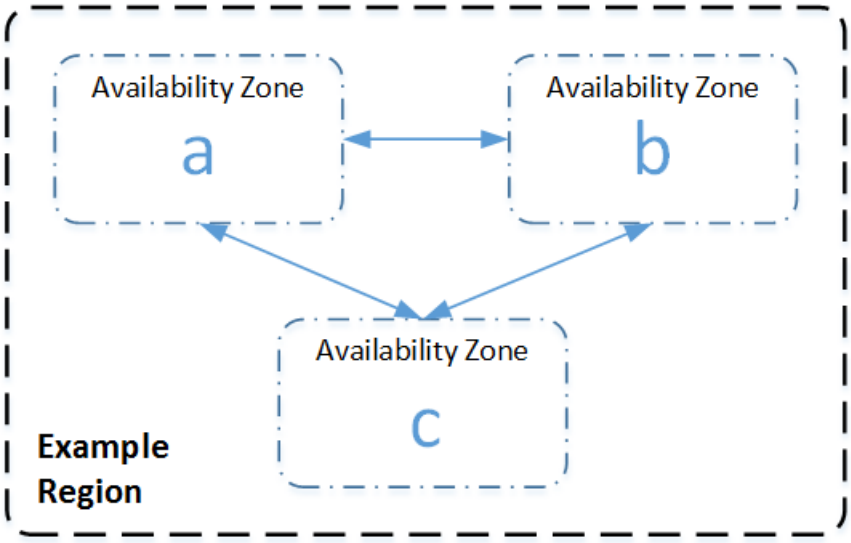
```
                "availability"
            ]
        },
        {
            "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
            "Message": "instance restarted",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:25:11.441Z",
            "SourceIdentifier": "docdb-2018-12-11-21-08-23",
            "EventCategories": [
                "availability"
            ]
        }
    ]
}
```

Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.

Auswählen von Regionen und Availability Zones

Amazon Cloud Computing-Ressourcen werden an mehreren Standorten weltweit gehostet. Diese Standorte bestehen aus AWS-Regionen und Availability Zones. Jeder AWS-Region ist ein separater geografischer Bereich. Jede Region verfügt über mehrere isolierte Standorte, die als Availability Zones bezeichnet werden. Amazon DocumentDB bietet Ihnen die Möglichkeit, Ressourcen wie Instances und Daten an mehreren Standorten zu platzieren. Ressourcen werden nur dann über repliziertAWS-Regionen, wenn Sie dies ausdrücklich tun.

Amazon betreibt hochmoderne, hoch verfügbare Rechenzentren. In seltenen Fällen kann es aber zu Ausfällen kommen, die die Verfügbarkeit von Instances desselben Standorts beeinträchtigen. Wenn Sie Ihre gesamten Instances an einem einzelnen Standort hosten, der von einem Ausfall dieser Art betroffen ist, ist keine Ihrer Instances verfügbar. Das folgende Diagramm zeigt eine AWS-Region mit drei Availability Zones.



Beachten Sie, dass alle Regionen voneinander unabhängig sind. Jede Amazon DocumentDB-Aktivität, die Sie initiieren (z. B. das Erstellen von Instances oder das Auflisten verfügbarer Instances), wird nur in Ihrem aktuellen Standard- ausgeführtAWS-Region. Sie können die Standardregion in der Konsole ändern, indem Sie die Umgebungsvariable `EC2_REGION` festlegen. Oder Sie können sie überschreiben, indem Sie den Parameter `--region` in der AWS CLI verwenden. Weitere Informationen finden Sie unter Konfigurieren der AWS Command Line Interface, insbesondere in den Abschnitten zu Umgebungsvariablen und Befehlszeilenoptionen.

Wenn Sie einen Cluster mit der Amazon DocumentDB-Konsole erstellen und ein Replikat in einer anderen Availability Zone erstellen möchten, erstellt Amazon DocumentDB zwei Instances. Es erstellt die primäre Instance in einer Availability Zone und die Replikat-Instance in einer anderen Availability Zone. Das Cluster-Volumen wird immer über drei Availability Zones repliziert.

Um eine Amazon DocumentDB-Instance in einer bestimmten zu erstellen oder damit zu arbeitenAWS-Region, verwenden Sie den entsprechenden regionalen Service-Endpunkt.

Verfügbarkeit in Regionen

Amazon DocumentDB ist in den folgenden AWS Regionen verfügbar.

Von Amazon DocumentDB unterstützte Regionen

Name der Region	Region	Availability Zones (Datenverarbeitung)
USA Ost (Ohio)	`us-east-2`	3

Name der Region	Region	Availability Zones (Datenverarbeitung)
USA Ost (Nord-Virginia)	us-east-1	6
USA West (Oregon)	us-west-2	4
Südamerika (São Paulo)	sa-east-1	3
Asien-Pazifik (Hongkong)	ap-east-1	3
Asien-Pazifik (Hyderabad)	ap-south-2	3
Asien-Pazifik (Mumbai)	ap-south-1	3
Asien-Pazifik (Seoul)	ap-northeast-2	4
Asien-Pazifik (Singapur)	ap-southeast-1	3
Asien-Pazifik (Sydney)	ap-southeast-2	3
Asien-Pazifik (Tokio)	ap-northeast-1	3
Kanada (Zentral)	ca-central-1	3
Region China (Peking)	cn-north-1	3
China (Ningxia)	cn-northwest-1	3
Europa (Frankfurt)	eu-central-1	3
Europa (Irland)	eu-west-1	3
Europa (London)	eu-west-2	3

Name der Region	Region	Availability Zones (Datenverarbeitung)
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
AWS GovCloud (USA-West)	`us-gov-west-1`	3
AWS GovCloud (USA-Ost)	`us-gov-east-1`	3

Standardmäßig ist die Zeitzone für einen Amazon DocumentDB-Cluster Universal Time Coordinated (UTC).

Informationen zum Suchen der Verbindungsendpunkte für Cluster und Instances in einer bestimmten Region finden Sie unter Grundlegendes zu Amazon DocumentDB-Endpunkten.

Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen

Sie können die Amazon DocumentDB DocumentDB-Engine-Konfiguration verwalten, indem Sie Parameter in einer Cluster-Parametergruppe verwenden. Eine Cluster-Parametergruppe ist eine Sammlung von Amazon DocumentDB-Konfigurationswerten, die die Verwaltung der Parameter Ihrer Amazon DocumentDB-Cluster erleichtern. Cluster-Parametergruppen dienen als Container für Engine-Konfigurationswerte, die auf jede alle Instances im Cluster angewendet werden.

In diesem Abschnitt wird beschrieben, wie Cluster-Parametergruppen erstellt, angezeigt und geändert werden. Es wird auch gezeigt, wie Sie ermitteln können, welche Cluster-Parametergruppe einem bestimmten Cluster zugeordnet ist.

Themen

- Beschreibung der Amazon DocumentDB-Cluster-Parametergruppen
- Amazon DocumentDB-Cluster-Parametergruppen erstellen
- Amazon DocumentDB-Cluster-Parametergruppen ändern
- Ändern von Amazon DocumentDB-Clustern zur Verwendung benutzerdefinierter Cluster-Parametergruppen
- Amazon DocumentDB-Cluster-Parametergruppen kopieren

- Amazon DocumentDB-Cluster-Parametergruppen zurücksetzen
- Löschen von Amazon DocumentDB-Cluster-Parametergruppen
- Referenz zu den Amazon DocumentDB-Clusterparametern

Beschreibung der Amazon DocumentDB-Cluster-Parametergruppen

Eine `default` Cluster-Parametergruppe wird automatisch erstellt, wenn Sie den ersten Amazon DocumentDB-Cluster in einer neuen Region erstellen oder eine neue Engine verwenden. Nachfolgende Cluster, die in derselben Region erstellt werden und dieselbe Engine-Version haben, werden mit der `default` Cluster-Parametergruppe erstellt.

Themen

- Beschreibung der Details einer Amazon DocumentDB-Cluster-Parametergruppe
- Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters

Beschreibung der Details einer Amazon DocumentDB-Cluster-Parametergruppe

Um die Details einer bestimmten Cluster-Parametergruppe zu beschreiben, führen Sie über die AWS Management Console oder die AWS Command Line Interface (AWS CLI) die folgenden Schritte aus.

Using the AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) den Namen der Parametergruppe aus, deren Details Sie anzeigen möchten.

4. Auf der anschließend angezeigten Seite werden die Parameter der Parametergruppe, die letzte Aktivität und Tags angezeigt.

   - In Cluster parameters (Cluster-Parameter) werden Ihnen der Name, der aktuelle Wert, die zulässigen Werte, der Anwendungstyp, der Datentyp und die Beschreibung des Parameters angezeigt. Sie können auch erkennen, ob der Parameter geändert werden kann. Sie können einzelne Parameter ändern, indem Sie den Parameter auswählen und dann im Abschnitt Cluster-Parameter auf Bearbeiten klicken. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parameter ändern.
   - In Recent events (Aktuelle Ereignisse) werden Ihnen die jeweils aktuellen Ereignisse für diese Parametergruppe angezeigt. Sie können diese Ereignisse über die Suchleiste in diesem Abschnitt filtern. Weitere Informationen finden Sie unter Verwalten von Amazon DocumentDB DocumentDB-Ereignissen.
   - Unter Tags werden die Tags angezeigt, die sich in dieser Cluster-Parametergruppe befinden. Sie können Tags hinzufügen oder entfernen, indem Sie im Abschnitt Tags die Option Bearbeiten wählen. Weitere Informationen finden Sie unter Taggen von Amazon DocumentDB-Ressourcen.

Using the AWS CLI

Sie können den `describe-db-cluster-parameter-groups` AWS CLI Befehl verwenden, um den Amazon-Ressourcennamen (ARN), die Familie, die Beschreibung und den Namen einer einzelnen Cluster-Parametergruppe oder aller Cluster-Parametergruppen, die Sie für Amazon DocumentDB haben, anzuzeigen. Sie können auch den AWS CLI-Befehl `describe-db-cluster-parameters` verwenden, um die Parameter und ihre Details innerhalb einer einzelnen Cluster-Parametergruppe anzuzeigen.

- **`--describe-db-cluster-parameter-groups`**— Um eine Liste all Ihrer Cluster-Parametergruppen und ihrer Details zu sehen.
  - **`--db-cluster-parameter-group-name`**— Fakultativ. Der Name der Cluster-Parametergruppe, die beschrieben werden soll. Wenn dieser Parameter weggelassen wird, werden alle Cluster-Parametergruppen beschrieben.
- **`--describe-db-cluster-parameters`**— Um alle Parameter innerhalb einer Parametergruppe und ihre Werte aufzulisten.
  - **`--db-cluster-parameter-group name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die beschrieben werden soll.

Example

Der folgende Code listet bis zu 100 Cluster-Parametergruppen zusammen mit ARN, Familie, Beschreibung und Name auf.

```
aws docdb describe-db-cluster-parameter-groups
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
          "DBClusterParameterGroups": [
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:012345678912:cluster-pg:default.docdb4.0",
                  "DBParameterGroupFamily": "docdb4.0",
                  "Description": "Default cluster parameter group for docdb4.0",
                  "DBClusterParameterGroupName": "default.docdb4.0"
              },
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:012345678912:cluster-pg:sample-parameter-group",
                  "DBParameterGroupFamily": "docdb4.0",
                  "Description": "Custom docdb4.0 parameter group",
                  "DBClusterParameterGroupName": "sample-parameter-group"
              }
          ]
}
```

Example

Der folgende Code listet den ARN, die Familie, die Beschreibung und den Namen für `sample-parameter-group` auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
          --db-cluster-parameter-group-name sample-parameter-group
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
```

```
        --db-cluster-parameter-group-name sample-parameter-group
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
        "DBClusterParameterGroups": [
            {
                "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:sample-parameter-group",
                "Description": "Custom docdb4.0 parameter group",
                "DBParameterGroupFamily": "docdb4.0",
                "DBClusterParameterGroupName": "sample-parameter-group"
            }
        ]
}
```

Example

Der folgende Code listet die Werte der Parameter in `sample-parameter-group` auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

Für Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
   "Parameters": [
        {
           "ParameterName": "audit_logs",
```

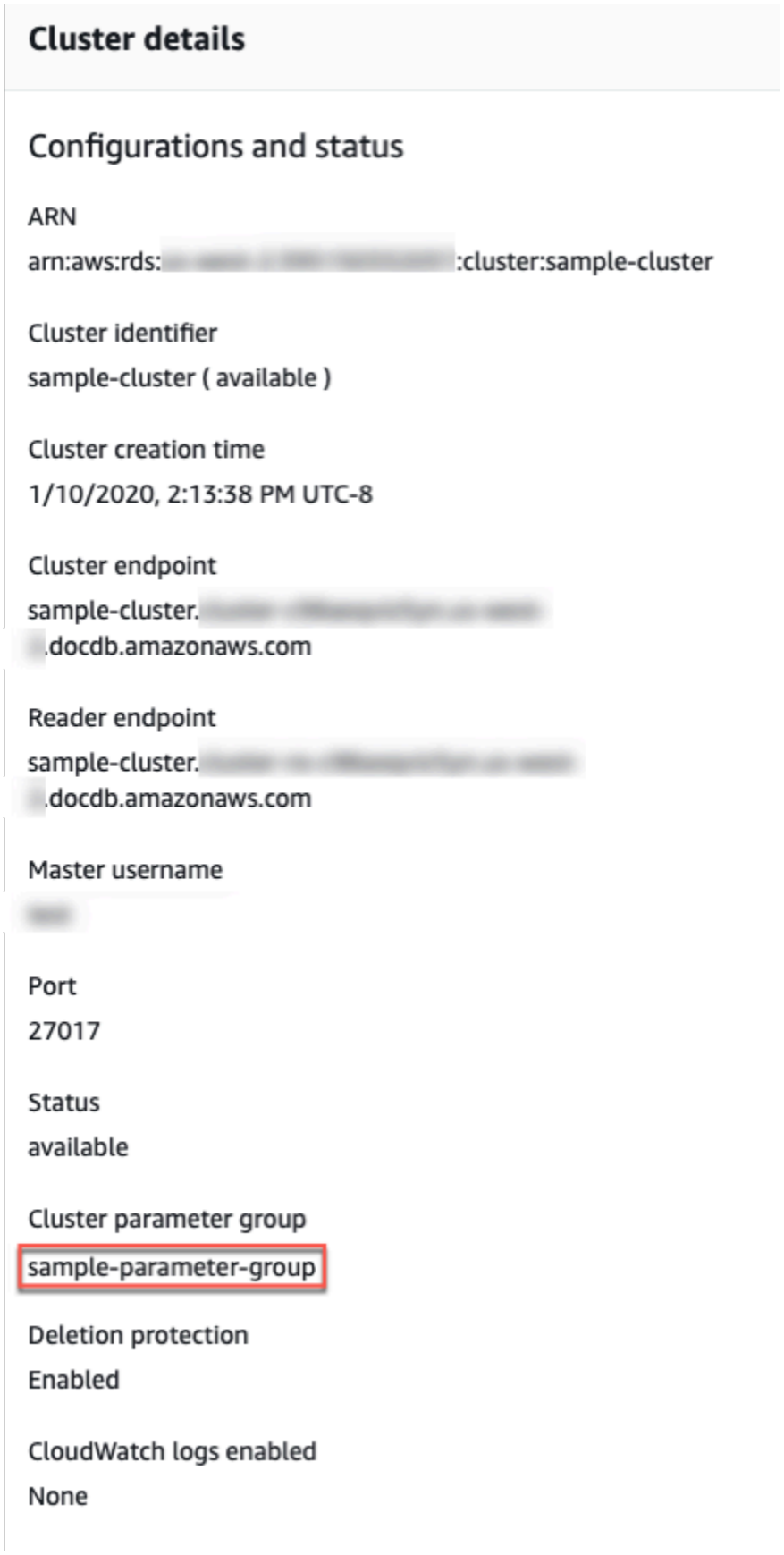
```
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
         },
         {
            "ParameterName": "change_stream_log_retention_duration",
            "ParameterValue": "17777",
            "Description": "Duration of time in seconds that the change stream log
 is retained and can be consumed.",
            "Source": "user",
            "ApplyType": "dynamic",
            "DataType": "integer",
            "AllowedValues": "3600-86400",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
         }
   ]
}
```


Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters

Um zu bestimmen, welche Parametergruppe einem bestimmten Cluster zugeordnet ist, führen Sie über die AWS Management Console oder die AWS CLI die folgenden Schritte aus.

Using the AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
3. Wählen Sie aus der Liste der Cluster den Namen des Clusters aus, an dem Sie interessiert sind.
4. Auf der anschließend angezeigten Seite werden die Details des von Ihnen ausgewählten Clusters angezeigt. Scrollen Sie nach unten zu Cluster details (Clusterdetails). Sie finden den Namen der Parametergruppe unten in diesem Abschnitt unterhalb von Cluster parameter group (Cluster-Parametergruppe).


Cluster details
Configurations and status
ARN
arn:aws:rds:
:cluster:sample-cluster
Cluster identifier
sample-cluster (available)
Cluster creation time
1/10/2020, 2:13:38 PM UTC-8
Cluster endpoint
sample-cluster.
.docdb.amazonaws.com
Reader endpoint
sample-cluster.
.docdb.amazonaws.com
Master username
Port
27017
Status
available
Cluster parameter group
sample-parameter-group
Deletion protection
Enabled
CloudWatch logs enabled
None

Using the AWS CLI

Der folgende AWS CLI-Code ermittelt die Parametergruppe des `sample-cluster`-Clusters.

```
aws docdb describe-db-clusters \
      --db-cluster-identifier sample-cluster \
      --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
       [
           "sample-cluster",
           "sample-parameter-group"
       ]
]
```

Amazon DocumentDB-Cluster-Parametergruppen erstellen

Standard-Cluster-Parametergruppen wie `default.docdb5.0default.docdb4.0`, oder werden erstellt`default.docdb3.6`, wenn Sie einen Cluster mit einer neuen Engine-Version und in einer neuen Region erstellen. Nachfolgende Cluster, die in dieser Region und mit derselben Engine-Version erstellt wurden, erben die `default` Cluster-Parametergruppe. Nach der Erstellung können die `default` Parametergruppen nicht gelöscht oder umbenannt werden. Sie können das Engine-Verhalten von Cluster-Instances ändern, indem Sie eine benutzerdefinierte Parametergruppe mit bevorzugten Parameterwerten erstellen und sie an Ihren Amazon DocumentDB-Cluster anhängen.

Das folgende Verfahren führt Sie durch die Erstellung einer benutzerdefinierten Cluster-Parametergruppe. Anschließend können Sie die Parameter innerhalb dieser Parametergruppe ändern.

> **Note**
>
> Sie sollten nach der Erstellung einer Cluster-Parametergruppe mindestens 5 Minuten warten, bevor Sie diese Cluster-Parametergruppe verwenden. Dadurch kann Amazon DocumentDB die `create` Aktion vollständig abschließen, bevor die Cluster-Parametergruppe für einen neuen Cluster verwendet wird. Sie können die AWS Management Console oder die AWS CLI-Operation `describe-db-cluster-parameter-groups` verwenden, um zu

überprüfen, ob die Cluster-Parametergruppe erstellt wurde. Weitere Informationen finden Sie unter Beschreibung der Amazon DocumentDB-Cluster-Parametergruppen.

Using the AWS Management Console

So erstellen Sie eine Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) die Option Create (Erstellen) aus.
4. Geben Sie im Bereich Create cluster parameter group (Cluster-Parametergruppe erstellen) Folgendes ein:

   a. Gruppenname — Geben Sie einen Namen für die Cluster-Parametergruppe ein. Beispiel: `sample-parameter-group` Für Cluster-Parametergruppen gelten die folgenden Benennungseinschränkungen:

      - Die Länge muss [1 bis 255] alphanumerische Zeichen betragen.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

   b. Beschreibung — Geben Sie eine Beschreibung für diese Cluster-Parametergruppe ein.
5. Um die Cluster-Parametergruppe zu erstellen, wählen Sie Erstellen aus. Um die Operation abzubrechen, wählen Sie Abbrechen aus.
6. Nach der Auswahl von Create (Erstellen) wird oben auf der Seite der folgende Text angezeigt, um zu bestätigen, dass die Cluster-Parametergruppe erfolgreich erstellt wurde:

```
Successfully created cluster parameter group 'sample-parameter-group'.
```

Using the AWS CLI

Um eine neue Cluster-Parametergruppe für Amazon DocumentDB 4.0-Cluster zu erstellen, verwenden Sie den AWS CLI `create-db-cluster-parameter-group` Vorgang mit den folgenden Parametern:

- **`--db-cluster-parameter-group-name`**— Der Name der benutzerdefinierten Cluster-Parametergruppe. Beispiel: `sample-parameter-group`
- **`--db-cluster-parameter-group-family`**— Die Cluster-Parametergruppen-Familie, die als Vorlage für die benutzerdefinierte Cluster-Parametergruppe verwendet wird. Derzeit muss diese `docdb4.0` lauten.
- **`--description`**— Die vom Benutzer bereitgestellte Beschreibung für diese Cluster-Parametergruppe. Das folgende Beispiel verwendet "`Custom docdb4.0 parameter group`“.

Für Linux, macOS oder Unix:

Example

```
aws docdb create-db-cluster-parameter-group \
 --db-cluster-parameter-group-name sample-parameter-group \
 --db-parameter-group-family docdb4.0 \
 --description "Custom docdb4.0 parameter group"
```

Für Windows:

```
aws docdb create-db-cluster-parameter-group ^
 --db-cluster-parameter-group-name sample-parameter-group ^
 --db-parameter-group-family docdb4.0 ^
 --description "Custom docdb4.0 parameter group"
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
```

```
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "sample-parameter-group",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Custom docdb4.0 parameter group",
        "DBClusterParameterGroupArn": "sample-parameter-group-arn"
    }
}
```

Amazon DocumentDB-Cluster-Parametergruppen ändern

In diesem Abschnitt wird erklärt, wie Sie eine benutzerdefinierte Amazon DocumentDB DocumentDB-Parametergruppe ändern. In Amazon DocumentDB können Sie eine `default` Cluster-Parametergruppe nicht ändern, die erstellt wird, wenn Sie zum ersten Mal einen Cluster mit einer neuen Engine-Version in einer neuen Region erstellen. Wenn Ihr Amazon DocumentDB-Cluster die Standard-Cluster-Parametergruppe verwendet und Sie einen Wert darin ändern möchten, müssen Sie zuerst eine neue Parametergruppe erstellen oder eine bestehende Parametergruppe kopieren, sie ändern und dann die geänderte Parametergruppe auf Ihren Cluster anwenden.

Gehen Sie wie folgt vor, um eine benutzerdefinierte Cluster-Parametergruppe zu ändern. Die Übertragung von Änderungsaktionen kann eine Weile dauern. Bitte warten Sie, bis die geänderte Cluster-Parametergruppe verfügbar ist, bevor Sie sie an Ihren Cluster anhängen. Sie können die AWS Management Console oder die AWS CLI-Operation `describe-db-cluster-parameters` verwenden, um zu überprüfen, ob die Cluster-Parametergruppe geändert wurde. Weitere Informationen finden Sie unter Beschreibung von Cluster-Parametergruppen.

Using the AWS Management Console

Gehen Sie wie folgt vor, um eine benutzerdefinierte Amazon DocumentDB DocumentDB-Parametergruppe zu ändern. Sie können eine `default`-Parametergruppe nicht ändern. Wenn Sie einen Wert in der `default`-Parametergruppe ändern möchten, können Sie die standardmäßige Cluster-Parametergruppe kopieren, sie ändern und die geänderte Parametergruppe anschließend auf Ihren Cluster anwenden. Weitere Hinweise zum Anwenden von Parametergruppen auf Ihren Cluster finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

So ändern Sie eine benutzerdefinierte Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Parameter groups (Parametergruppen) aus. Wählen Sie in der Liste der Parametergruppen den Namen der Parametergruppe, die Sie ändern möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Gehen Sie bei jedem Parameter in der Parametergruppe, die Sie ändern möchten, wie folgt vor:

   a. Suchen Sie den Parameter, den Sie ändern möchten, und überprüfen Sie, ob er änderbar ist, indem Sie überprüfen, ob er in der `true` Spalte Modifizierbar aufgeführt ist.
   b. Wenn der Parameter geändert werden kann, wählen Sie ihn aus und wählen oben rechts auf der Konsolenseite Edit (Bearbeiten) aus.
   c. Nehmen Sie im Dialogfeld Modify **<parameter-name>** die gewünschten Änderungen vor. Wählen Sie anschließend Modify cluster parameter (Cluster-Parameter ändern) aus. Um die Änderungen zu verwerfen, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Sie können den `ParameterValueDescription`, oder `ApplyMethod` eines beliebigen modifizierbaren Parameters in einer benutzerdefinierten Amazon DocumentDB-Cluster-Parametergruppe mit dem ändern. AWS CLI Sie können eine standardmäßige Cluster-Parametergruppe nicht direkt ändern.

Um die Parameter einer benutzerdefinierten Cluster-Parametergruppe zu ändern, verwenden Sie die Operation `modify-db-cluster-parameter-group` mit den folgenden Parametern.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die Sie ändern.
- **`--parameters`** – Erforderlich. Die Parameter, die von Ihnen geändert werden. Eine Liste der Parameter, die für alle Instances in einem Amazon DocumentDB-Cluster gelten, finden Sie unter. Referenz zu den Amazon DocumentDB-Clusterparametern Jeder Parametereintrag muss Folgendes enthalten:

- **`ParameterName`**— Der Name des Parameters, den Sie ändern.
- **`ParameterValue`**— Der neue Wert für diesen Parameter.
- **`ApplyMethod`**— Wie die Änderungen an diesem Parameter angewendet werden sollen. Zugelassene Werte sind `immediate` und `pending-reboot`.

> **Note**
>
> Parameter mit dem `ApplyType` von `static` müssen über einen `ApplyMethod` von `pending-reboot` verfügen.

Example - Ändern eines Parameterwerts

In diesem Beispiel listen Sie die Parameterwerte von `sample-parameter-group` auf und ändern den `tls`-Parameter. Anschließend listen Sie nach einer Wartezeit von 5 Minuten die Parameterwerte von `sample-parameter-group` erneut auf, um die geänderten Parameterwerte anzuzeigen.

1. Listen Sie die Parameter von `sample-parameter-group` und ihre Werte auf.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-cluster-parameters \
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   Für Windows:

   ```
   aws docdb describe-db-cluster-parameters ^
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

   ```
   {
          "Parameters": [
              {
                  "Source": "system",
                  "ApplyType": "static",
                  "AllowedValues": "disabled,enabled",
                  "ParameterValue": "enabled",
   ```

```
                "ApplyMethod": "pending-reboot",
                "DataType": "string",
                "ParameterName": "tls",
                "IsModifiable": true,
                "Description": "Config to enable/disable TLS"
            },
            {
                "Source": "user",
                "ApplyType": "dynamic",
                "AllowedValues": "disabled,enabled",
                "ParameterValue": "enabled",
                "ApplyMethod": "pending-reboot",
                "DataType": "string",
                "ParameterName": "ttl_monitor",
                "IsModifiable": true,
                "Description": "Enables TTL Monitoring"
            }
        ]
}
```

2. Ändern Sie den Parameter `tls`, sodass der Wert `disabled` ist.

   Sie können den `ApplyMethod` nicht ändern, da der `ApplyType` `static` ist.

   Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --parameters
 "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
```

   Für Windows:

```
aws docdb modify-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --parameters
 "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
       "DBClusterParameterGroupName": "sample-parameter-group"
```

```
    }
```

3. Warten Sie mindestens 5 Minuten.
4. Listen Sie die Parameterwerte von `sample-parameter-group` auf, um zu überprüfen, ob der Parameter `tls` geändert wurde.

   Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
```

   Für Windows:

```
aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
       "Parameters": [
           {
               "ParameterValue": "false",
               "ParameterName": "enable_audit_logs",
               "ApplyType": "dynamic",
               "DataType": "string",
               "Description": "Enables auditing on cluster.",
               "AllowedValues": "true,false",
               "Source": "system",
               "IsModifiable": true,
               "ApplyMethod": "pending-reboot"
           },
           {
               "ParameterValue": "disabled",
               "ParameterName": "tls",
               "ApplyType": "static",
               "DataType": "string",
               "Description": "Config to enable/disable TLS",
               "AllowedValues": "disabled,enabled",
               "Source": "system",
               "IsModifiable": true,
               "ApplyMethod": "pending-reboot"
```

```
            }
        ]
}
```

Ändern von Amazon DocumentDB-Clustern zur Verwendung benutzerdefinierter Cluster-Parametergruppen

Wenn Sie einen Amazon DocumentDB-Cluster erstellen, wird automatisch eine `default.docdb4.0` Parametergruppe für diesen Cluster erstellt. Sie können die Cluster-Parametergruppe `default` nicht ändern. Stattdessen können Sie Ihren Amazon DocumentDB-Cluster ändern, um ihm eine neue benutzerdefinierte Parametergruppe zuzuordnen.

In diesem Abschnitt wird erklärt, wie ein vorhandener Amazon DocumentDB-Cluster so geändert wird, dass er eine benutzerdefinierte Cluster-Parametergruppe mit AWS Management Console und AWS Command Line Interface (AWS CLI) verwendet.

Using the AWS Management Console

Um einen Amazon DocumentDB-Cluster so zu ändern, dass er eine neue, nicht standardmäßige Cluster-Parametergruppe verwendet

1. Bevor Sie beginnen, stellen Sie sicher, dass Sie einen Amazon DocumentDB-Cluster und eine Cluster-Parametergruppe erstellt haben. Weitere Anweisungen finden Sie unter Erstellen eines Amazon DocumentDB-Clusters und Amazon DocumentDB-Cluster-Parametergruppen erstellen.
2. Nachdem Sie Ihre Cluster-Parametergruppe erstellt haben, öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb. Wählen Sie im Navigationsbereich Clusters (Cluster) aus, um die neue Parametergruppe einem Cluster hinzuzufügen.
3. Wählen Sie den Cluster aus, dem Sie die Parametergruppe zuordnen möchten. Wählen Sie Actions (Aktionen= und anschließend Modify (Ändern) aus, um den Cluster zu ändern.
4. Wählen Sie unter Cluster options (Clusteroptionen) die neue Parametergruppe aus, der Sie den Cluster zuordnen möchten.
5. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.

6. Nachdem Sie Ihre Änderungen überprüft haben, können Sie diese sofort oder während des nächsten Wartungsfensters unter Scheduling of modifications (Planen von Änderungen) anwenden.
7. Wählen Sie Modify cluster (Cluster ändern) aus, um den Cluster mit der neuen Parametergruppe zu aktualisieren.

Using the AWS CLI

Bevor Sie beginnen, stellen Sie sicher, dass Sie einen Amazon DocumentDB-Cluster und eine Cluster-Parametergruppe erstellt haben. Mit diesem AWS CLI `create-db-cluster` Vorgang können Sie einen Amazon DocumentDB-Cluster erstellen. Sie können mithilfe des AWS CLI `create-db-cluster-parameter-group` Vorgangs eine Cluster-Parametergruppe erstellen.

Um dem Cluster die neue Cluster-Parametergruppe hinzuzufügen, verwenden Sie die AWS CLI-Operation `modify-db-cluster` mit den folgenden Parametern.

- -- db-cluster-identifier — Der Name Ihres Clusters (zum Beispiel`sample-cluster`).
- -- db-cluster-parameter-group -name — Der Name der Parametergruppe, der Sie Ihren Cluster zuordnen möchten (z. B.`sample-parameter-group`).

Example

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster
    --db-cluster-parameter-group-name sample-parameter-group
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
"DBCluster": {
    "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster",
    "DBClusterParameterGroup": "sample-parameter-group",
    "DBSubnetGroup": "default",
```

```
    ...
}
```

Amazon DocumentDB-Cluster-Parametergruppen kopieren

Sie können eine Kopie einer Cluster-Parametergruppe in Amazon DocumentDB mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) erstellen.

Using the AWS Management Console

Das folgende Verfahren führt Sie durch die Erstellung einer neuen Cluster-Parametergruppe, indem Sie eine vorhandene Cluster-Parametergruppe kopieren.

So kopieren Sie eine Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) den Namen der Cluster-Parametergruppe aus, die Sie kopieren möchten.
4. Wählen Sie Actions (Aktionen) und anschließend Copy (Kopieren) aus, um diese Parametergruppe zu kopieren.
5. Geben Sie in Copy options (Kopieroptionen) einen Namen und eine Beschreibung für die neue Cluster-Parametergruppe ein. Wählen Sie anschließend Copy (Kopieren) aus, um Ihre Änderungen zu speichern.

Using the AWS CLI

Um eine Kopie einer Cluster-Parametergruppe zu erstellen, verwenden Sie die `copy-db-cluster-parameter-group`-Operation mit den folgenden Parametern.

- **`--source-db-cluster-parameter-group-identifier`** – Erforderlich. Der Name oder Amazon-Ressourcenname (ARN) der Cluster-Parametergruppe, von der Sie eine Kopie erstellen möchten.

  Wenn sich die Quell- und Zielcluster-Parametergruppen in derselben Gruppe befindenAWS-Region, kann der Identifier entweder ein Name oder ein ARN sein.

Wenn sich die Quell- und Zielcluster-Parametergruppen unterscheidenAWS-Regionen, muss es sich bei der ID um einen ARN handeln.

- **`--target-db-cluster-parameter-group-identifier`** – Erforderlich. Der Name oder ARN der Kopie der Cluster-Parametergruppe.

  Einschränkungen:

  - Kann nicht Null, leer oder negativ sein.
  - Muss 1—255 Buchstaben, Zahlen oder Bindestriche enthalten.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- **`--target-db-cluster-parameter-group-description`** – Erforderlich. Eine vom Benutzer angegebene Beschreibung für die Kopie der Cluster-Parametergruppe.

Example

Der folgende Code erstellt eine Kopie von `sample-parameter-group` und benennt die Kopie `sample-parameter-group-copy`.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier sample-parameter-group \
    --target-db-cluster-parameter-group-identifier sample-parameter-group-copy \
    --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

Für Windows:

```
aws docdb copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier sample-parameter-group ^
    --target-db-cluster-parameter-group-identifier sample-parameter-group-copy ^
    --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBClusterParameterGroup": {
```

```
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:sample-parameter-group-copy",
        "DBClusterParameterGroupName": "sample-parameter-group-copy",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Copy of sample-parameter-group"
    }
}
```

Amazon DocumentDB-Cluster-Parametergruppen zurücksetzen

Sie können einige oder alle Parameterwerte einer Amazon DocumentDB-Cluster-Parametergruppe auf ihre Standardwerte zurücksetzen, indem Sie das AWS Management Console oder das AWS Command Line Interface (AWS CLI) verwenden, um die Cluster-Parametergruppe zurückzusetzen.

Using the AWS Management Console

Führen Sie die folgenden Schritt aus, um einige oder alle Parameterwerte einer Cluster-Parametergruppe auf die Standardwerte zurückzusetzen.

So setzen Sie die Parameterwerte einer Cluster-Parametergruppe zurück

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Parameter groups (Parametergruppen) aus.
3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) den Namen der Cluster-Parametergruppe aus, die Sie zurücksetzen möchten.
4. Wählen Sie Actions (Aktionen) und anschließend Reset (Zurücksetzen) aus, um diese Parametergruppe zurückzusetzen.
5. Bestätigen Sie auf der Seite Cluster parameter group reset confirmation (Zurücksetzen der Cluster-Parametergruppe bestätigen), dass Sie alle Cluster-Parameter für diese Parametergruppe auf die Standardwerte zurücksetzen möchten. Wählen Sie anschließend Reset (Zurücksetzen) aus, um die Parametergruppe zurückzusetzen. Sie können auch Cancel (Abbrechen) auswählen, um die Änderungen zu verwerfen.

Using the AWS CLI

Um einige oder alle Parameterwerte einer Cluster-Parametergruppe auf ihre Standardwerte zurückzusetzen, verwenden Sie die `reset-db-cluster-parameter-group`-Operation mit den folgenden Parametern.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der zurückzusetzenden Cluster-Parametergruppe.
- **`--parameters`**— Fakultativ. Eine Liste von `ParameterName` und `ApplyMethod` in der Cluster-Parametergruppe, die auf ihre Standardwerte zurücksetzt werden sollen. Statische Parameter müssen auf `pending-reboot` gesetzt sein, damit sie beim nächsten Neustart der Instance oder `reboot-db-instance`-Anforderung wirksam werden. Sie müssen `reboot-db-instance` für alle Instances in Ihrem Cluster aufrufen, auf die der aktualisierte statische Parameter angewendet werden soll.

  Dieser Parameter und `--reset-all-parameters` schließen sich gegenseitig aus. Sie können jede der beiden Optionen verwenden, jedoch nicht beide.
- **`--reset-all-parameters`**oder **`--no-reset-all-parameters`** — Fakultativ. Gibt an, ob alle Parameter (`--reset-all-parameters`) oder nur einige der Parameter (`--no-reset-all-parameters`) auf ihre Standardwerte zurückgesetzt werden sollen. Der `--reset-all-parameters`-Parameter und `--parameters` schließen sich gegenseitig aus. Sie können jede der beiden Optionen verwenden, jedoch nicht beide.

  Wenn Sie die gesamte Gruppe zurücksetzen, werden dynamische Parameter sofort aktualisiert. Statische Parameter werden auf `pending-reboot` festgelegt, damit sie beim nächsten Neustart der Instance oder bei der nächsten `reboot-db-instance`-Anforderung wirksam werden. Sie müssen `reboot-db-instance` für alle Instances in Ihrem Cluster aufrufen, auf die der aktualisierte statische Parameter angewendet werden soll.

Example

Beispiel 1: Zurücksetzen aller Parameter auf ihre Standardwerte

Der folgende Code setzt alle Parameter in der Cluster-Parametergruppe `sample-parameter-group` auf ihre Standardwerte zurück.

Für Linux, macOS oder Unix:

```
aws docdb reset-db-cluster-parameter-group \
```

```
        --db-cluster-parameter-group-name sample-parameter-group \
        --reset-all-parameters
```

Für Windows:

```
aws docdb reset-db-cluster-parameter-group ^
        --db-cluster-parameter-group-name sample-parameter-group ^
        --reset-all-parameters
```

Beispiel 2: Zurücksetzen von angegebenen Parametern auf ihre Standardwerte

Der folgende Code setzt den `tls`-Parameter in der Cluster-Parametergruppe `sample-parameter-group` auf seinen Standardwert zurück.

Für Linux, macOS oder Unix:

```
aws docdb reset-db-cluster-parameter-group \
        --db-cluster-parameter-group-name sample-parameter-group \
        --no-reset-all-parameters \
        --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

Für Windows:

```
aws docdb reset-db-cluster-parameter-group ^
        --db-cluster-parameter-group-name sample-parameter-group ^
        --no-reset-all-parameters ^
        --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
        "DBClusterParameterGroupName": "sample-parameter-group"
   }
```

Neustarten einer Cluster-Instance

Bevor ein statischer Parameterwert geändert wird, muss die Cluster-Instance neu gestartet werden. Starten Sie alle Instances in Ihrem Cluster neu, auf die der aktualisierte statische Parameter angewendet werden soll.

Für Linux, macOS oder Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-cluster-instance
```

Für Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-cluster-instance
```

Löschen von Amazon DocumentDB-Cluster-Parametergruppen

Sie können eine benutzerdefinierte Amazon DocumentDB-Cluster-Parametergruppe mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) löschen. Sie können die Cluster-Parametergruppe `default.docdb4.0` nicht löschen.

Using the AWS Management Console

So löschen Sie eine Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parametergruppen das Optionsfeld links neben der Cluster-Parametergruppe aus, die Sie löschen möchten.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie im Bestätigungsbereich Delete (Löschen) die Option Delete (Löschen) aus, um die Cluster-Parametergruppe zu löschen. Um die Cluster-Parametergruppe beizubehalten, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Um eine Cluster-Parametergruppe zu löschen, verwenden Sie die `delete-db-cluster-parameter-group`-Operation mit dem folgenden Parameter.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der zu löschenden Cluster-Parametergruppe. Dabei muss es sich um eine vorhandene Cluster-Parametergruppe handeln. Sie können die Cluster-Parametergruppe *`default.docdb4.0`* nicht löschen.

Example - Löschen einer Cluster-Parametergruppe

Das folgende Beispiel führt Sie durch die drei Schritte zum Löschen einer Cluster-Parametergruppe:

1. Ermitteln des Namens der Cluster-Parametergruppe, die Sie löschen möchten.
2. Löschen der angegebene Cluster-Parametergruppe.
3. Überprüfen, ob die Cluster-Parametergruppe gelöscht wurde.

1. Suchen Sie den Namen der Cluster-Parametergruppe, die Sie löschen möchten.

Der folgende Code listet die Namen aller Cluster-Parametergruppen auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Die Ausgabe der vorherigen Operation ist eine Liste der Namen der Cluster-Parametergruppen ähnlich der folgenden (JSON-Format).

```
[
       [
           "default.docdb4.0"
       ],
```

```
        [
            "sample-parameter-group"
        ],
        [
            "sample-parameter-group-copy"
        ]
    ]
```

2. Löschen Sie eine angegebene Cluster-Parametergruppe.

Der folgenden Code löscht die Cluster-Parametergruppe `sample-parameter-group-copy`.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Für Windows:

```
aws docdb delete-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Diese Operation erzeugt keine Ausgabe.

3. Überprüfen Sie, ob die angegebene Cluster-Parametergruppe gelöscht wurde.

Der folgende Code listet die Namen aller verbleibenden Cluster-Parametergruppen auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Die Ausgabe der obigen Operation ist eine Liste der Cluster-Parametergruppen, die der folgenden ähnelt (JSON-Format). Die Cluster-Parametergruppe, die Sie gerade gelöscht haben, sollte nicht in der Liste enthalten sein.

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
       [
           "default.docdb4.0"
       ],
       [
           "sample-parameter-group"
       ]
   ]
```

Referenz zu den Amazon DocumentDB-Clusterparametern

Wenn Sie einen dynamischen Parameter ändern und die Cluster-Parametergruppe speichern, wird diese Änderung sofort übernommen, ungeachtet der Einstellung von Apply immediately (Sofort anwenden). Wenn Sie einen statischen Parameter ändern und eine Cluster-Parametergruppe speichern, wird die Änderung des Parameters nach einem manuellen Neustart der Instance angewendet. Sie können eine Instance über die Amazon DocumentDB DocumentDB-Konsole oder durch expliziten Aufruf `reboot-db-instance` neu starten.

Die folgende Tabelle zeigt die Parameter, die für alle Instances in einem Amazon DocumentDB-Cluster gelten.

Amazon DocumentDB DocumentDB-Parameter auf Clusterebene

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`audit_logs`	disabled	aktiviert, deaktiviert, ddl, dml_read, dml_write, all, none	Ja	Dynamisch	String	Definiert, ob CloudWatch Amazon-Audit-Logs aktiviert sind. • **`enabled`**—CloudWatch Audit-

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						Logs sind aktiviert. • **disabled**— CloudWatch Audit-Logs sind deaktiviert. • **ddl**— Die Überwachung von DDL-Ereignissen ist aktiviert. • **dml_read**— Die Prüfung auf DML-Leseereignisse ist aktiviert. • **dml_write** — Die Überwachung von DML-Schre

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						ibereignissen ist aktiviert. • **all**— Das Auditing für alle Datenbankereignisse ist aktiviert. • **none**— Auditing ist deaktiviert.
`change_stream_log_retention_duration`	10800	3600-604800	Ja	Dynamisch	Ganzzahl	Definiert den Zeitraum (in Sekunden), den das Change Stream-Protokoll aufbewahrt und genutzt werden kann.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`profiler`	disabled	aktiviert, deaktiviert	Ja	Dynamisch	String	Aktiviert die Profilerstellung für langsame Operationen. • **`enabled`**— Vorgänge, die länger dauern als ein vom Kunden definierter Schwellenwert (z. B. 100 ms), werden in Amazon Logs protokolliert. CloudWatch • **`disabled`**— langsame Vorgänge werden

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						nicht in Logs protokolliert. CloudWatch
`profiler_sampling_rate`	1,0	0,0 – 1,0	Ja	Dynamisch	Gleitkommazahl	Definiert die Sampling-Rate für protokollierte Operationen.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`profiler_threshold_ms`	100	50 – 2147483646	Ja	Dynamisch	Ganzzahl	Definiert den Schwellenwert für `profiler`. • Alle Operationen, die größer als `profiler_threshold_ms` sind, werden in CloudWatch Logs protokolliert.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`tls`	aktiviert	aktiviert, deaktiviert, FIPS-140-3	Ja	Statisch	String	Gibt an, ob TLS-Verbindungen (Transport Layer Security) erforderlich sind. • **`enabled`**— Für die Verbindung sind TLS-Verbindungen erforderlich. • **`disabled`**— TLS-Verbindungen können nicht für die Verbindung verwendet werden. • **`fips-140-3`** — Für die Verbindun

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						g sind TLS-Verbindungen mit FIPS-Attributen (Federal Information Processing Standards) erforderlich. Der Cluster akzeptiert nur sichere Verbindungen gemäß FIPS-Publikation 140-3. Dies wird erst ab Amazon DocumentDB 5.0-Clustern (Engine-

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						Version 3.0.3727) in diesen Regionen unterstützt: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`ttl_monitor`	aktiviert	aktiviert, deaktiviert	Ja	Dynamisch	String	Gibt an, ob TTL-Überwachung (Time to Live) für den Cluster aktiviert ist. • **`enabled`**— Die TTL-Überwachung ist aktiviert. • **`disabled`**— Die TTL-Überwachung ist deaktiviert.

Amazon DocumentDB-Cluster-Parameter ändern

In Amazon DocumentDB bestehen Cluster-Parametergruppen aus Parametern, die für alle Instances gelten, die Sie im Cluster erstellen. Im Fall benutzerdefinierter Cluster-Parametergruppen können Sie für alle von Ihnen erstellten Parametergruppen die Parameterwerte jederzeit ändern oder alle Parameterwerte jederzeit auf die Standardwerte für Parametergruppen zurücksetzen. In diesem Abschnitt wird beschrieben, wie Sie die Parameter, aus denen eine Amazon DocumentDB-Cluster-Parametergruppe besteht, und ihre Werte anzeigen und wie Sie diese Werte ändern oder aktualisieren können.

Parameter können dynamisch oder statisch sein. Wenn Sie einen dynamischen Parameter ändern und die Cluster-Parametergruppe speichern, wird die Änderung sofort übernommen, ungeachtet der

Einstellung für `Apply Immediately`. Wenn Sie einen statischen Parameter ändern und die Cluster-Parametergruppe speichern, wird die Änderung des Parameters erst nach einem manuellen Neustart der Instances angewendet.

Parameter einer Amazon DocumentDB-Cluster-Parametergruppe anzeigen

Sie können die Parameter eines Amazon DocumentDB-Clusters und ihre Werte mit dem AWS Management Console oder AWS CLI sehen.

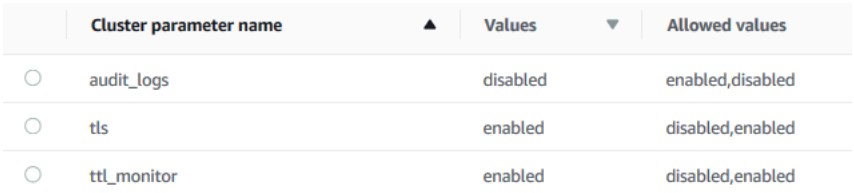
Using the AWS Management Console

So zeigen Sie die Details einer Cluster-Parametergruppe an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parameter groups (Parametergruppen) den Namen der Cluster-Parametergruppe aus, für die Sie die Details anzeigen möchten.
4. Auf der anschließend angezeigten Seite werden Ihnen für die einzelnen Parameter die folgenden Werte angezeigt: Name, aktueller Wert, zulässige Werte, ob der Parameter geändert werden kann, Anwendungstyp, Datentyp und Beschreibung.



Cluster parameter name	Values	Allowed values
audit_logs	disabled	enabled,disabled
tls	enabled	disabled,enabled
ttl_monitor	enabled	disabled,enabled

Using the AWS CLI

Um die Parameter einer Cluster-Parametergruppe und deren Werte anzuzeigen, verwenden Sie die `describe-db-cluster-parameters`-Operation mit den folgenden Parametern.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, für die Sie eine detaillierte Parameterliste möchten.
- **`--source`**— Fakultativ. Wenn angegeben, werden nur Parameter für eine bestimmte Quelle zurückgegeben. Parameterquellen können `engine-default`, `system` oder `user` sein.

Example

Der folgende Code listet die Parameter und deren Werte für die Parametergruppe `custom3-6-param-grp` auf. Um weitere Informationen zur Parametergruppe zu erhalten, lassen Sie die Zeile `--query` aus. Um Informationen zu allen Parametergruppen zu erhalten, lassen Sie die Zeile `--db-cluster-parameter-group-name` aus.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
   --db-cluster-parameter-group-name custom3-6-param-grp \
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

Für Windows:

```
aws docdb describe-db-cluster-parameters ^
   --db-cluster-parameter-group-name custom3-6-param-grp ^
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "audit_logs",
        "disabled"
    ],
    [
        "tls",
        "enabled"
    ],
    [
        "ttl_monitor",
        "enabled"
    ]
]
```

Parameter einer Amazon DocumentDB-Cluster-Parametergruppe ändern

Sie können die Parameter einer Parametergruppe über die AWS Management Console oder die AWS CLI ändern.

Using the AWS Management Console

So aktualisieren Sie die Parameter einer Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parameter groups (Parametergruppen) die Cluster-Parametergruppe aus, deren Parameter Sie aktualisieren möchten.
4. Auf der resultierenden Seite werden die Parameter und die entsprechenden Details zu dieser Cluster-Parametergruppe angezeigt. Wählen Sie einen zu aktualisierenden Parameter aus.
5. Klicken Sie oben rechts auf der Seite auf Edit (Bearbeiten), um den Wert des Parameters zu ändern. Weitere Informationen zu den Typen von Cluster-Parametern finden Sie unter Referenz zu den Amazon DocumentDB-Clusterparametern.
6. Führen Sie die Änderung aus. Wählen Sie anschließend Modify cluster parameter (Cluster-Parameter ändern) aus, um die Änderungen zu speichern. Um die Änderungen zu verwerfen, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Um die Parameter einer Cluster-Parametergruppe zu ändern, verwenden Sie die `modify-db-cluster-parameter-group`-Operation mit den folgenden Parametern:

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die Sie ändern.

- **`--parameters`** – Erforderlich. Der Parameter oder die Parameter, die Sie ändern. Jeder Parametereintrag muss Folgendes enthalten:
  - **`ParameterName`**— Der Name des Parameters, den Sie ändern.
  - **`ParameterValue`**— Der neue Wert für diesen Parameter.
  - **`ApplyMethod`**— Wie die Änderungen an diesem Parameter angewendet werden sollen. Zugelassene Werte sind `immediate` und `pending-reboot`.

    > **Note**
    >
    > Parameter mit dem `ApplyType` von `static` müssen über einen `ApplyMethod` von `pending-reboot` verfügen.

So ändern Sie die Werte der Parameter einer Cluster-Parametergruppe (AWS CLI)

Im folgenden Beispiel wird der Parameter `tls` geändert.

1. Listen Sie die Parameter von **`sample-parameter-group`** und ihre Werte auf.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Für Windows:

   ```
   aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

   ```
   {
       "Parameters": [
           {
               "Source": "system",
               "ApplyType": "static",
               "AllowedValues": "disabled,enabled",
               "ParameterValue": "enabled",
               "ApplyMethod": "pending-reboot",
   ```

```
            "DataType": "string",
            "ParameterName": "tls",
            "IsModifiable": true,
            "Description": "Config to enable/disable TLS"
        },
        {
            "Source": "user",
            "ApplyType": "dynamic",
            "AllowedValues": "disabled,enabled",
            "ParameterValue": "enabled",
            "ApplyMethod": "pending-reboot",
            "DataType": "string",
            "ParameterName": "ttl_monitor",
            "IsModifiable": true,
            "Description": "Enables TTL Monitoring"
        }
    ]
}
```

2. Ändern Sie den **`tls`**-Parameter, sodass sein Wert **`disabled`** ist. Sie können den `ApplyMethod` nicht ändern, da der `ApplyType` `static` ist.

   Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 "ParameterName"=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

   Für Windows:

```
aws docdb modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "DBClusterParameterGroupName": "sample-parameter-group"
}
```

3. Warten Sie mindestens 5 Minuten.
4. Listen Sie die Parameterwerte von **`sample-parameter-group`** auf.

   Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

   Für Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "Parameters": [
        {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "tls",
            "ParameterValue": "disabled",
            "Description": "Config to enable/disable TLS",
            "Source": "user",
            "ApplyType": "static",
            "DataType": "string",
            "AllowedValues": "disabled,enabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        }
    ]
}
```

Grundlegendes zu Amazon DocumentDB-Endpunkten

Sie können Amazon DocumentDB-Endpunkte (mit MongoDB-Kompatibilität) verwenden, um eine Verbindung zu einem Cluster oder einer Instance herzustellen. Amazon DocumentDB hat drei verschiedene Arten von Endpunkten, von denen jeder seinen eigenen Zweck hat.

Themen

Cluster-Endpunkt

Ein Cluster-Endpunkt ist ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zur aktuellen primären Instance für den Cluster herstellt. Jeder Amazon DocumentDB-Cluster hat einen einzelnen Cluster-Endpunkt und eine primäre Instance. Im Falle eines Failovers wird der Cluster-Endpunkt auf die neue primäre Instance umgeschaltet.

Leser-Endpunkt

Ein Reader-Endpunkt ist ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zu einem der verfügbaren Replikate für diesen Cluster herstellt. Jeder Amazon DocumentDB-Cluster hat einen Reader-Endpunkt. Wenn es mehr als ein Replikat gibt, leitet der Reader-Endpunkt jede Verbindungsanfrage an eines der Amazon DocumentDB-Replikate weiter.

Instance-Endpunkt

Ein Instance-Endpunkt ist ein Endpunkt, der sich mit einer bestimmten Instance verbindet. Jede Instance in einem Cluster, unabhängig davon, ob es sich um eine primäre oder eine Replikat-Instance handelt, hat ihren eigenen eindeutigen Instance-Endpunkt. Sie sollten keine Instance-Endpunkte in Ihrer Anwendung verwenden. Denn diese können im Falle eines Failovers Rollen ändern, sodass Code-Änderungen in Ihrer Anwendung erforderlich sind.

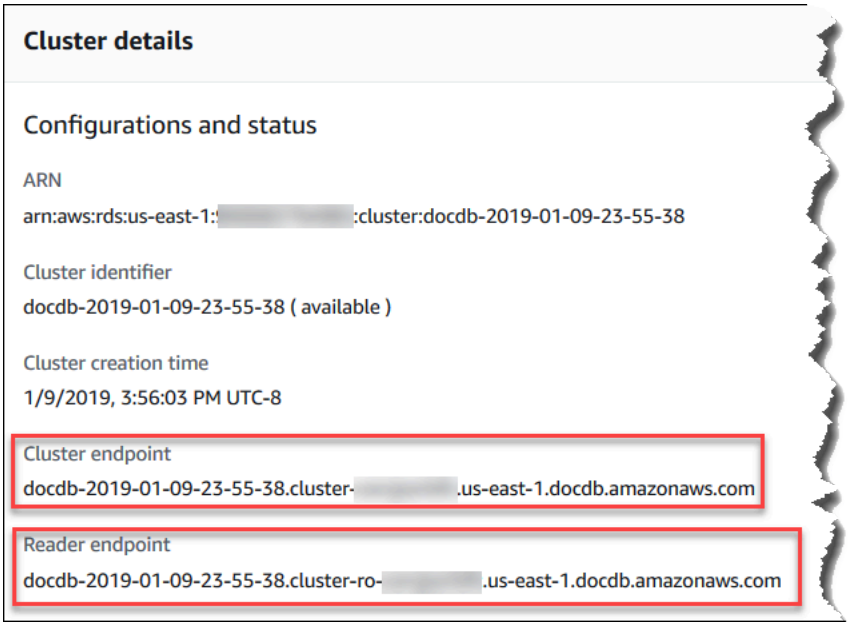
Suchen der Endpunkte eines Clusters

Sie können den Cluster-Endpunkt und den Reader-Endpunkt eines Clusters mithilfe der Amazon DocumentDB-Konsole finden oderAWS CLI.

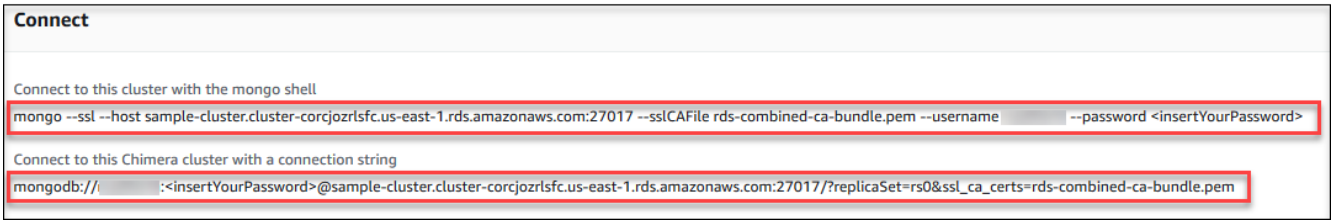
Using the AWS Management Console

So finden Sie die Endpunkte eines Clusters über die Konsole:

1. Melden Sie sich an beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich cluster aus.
3. Wählen Sie aus der Liste der Cluster den Namen des Clusters, der Sie interessiert.
4. Scrollen Sie nach unten zum Abschnitt Details und suchen Sie den Cluster-Endpunkt und den Reader-Endpunkt.



5. Um eine Verbindung zu diesem Cluster herzustellen, scrollen Sie nach oben zum Abschnitt Connect (Verbinden). Suchen Sie die Verbindungszeichenfolge für die mongo-Shell und eine Verbindungszeichenfolge, die im Anwendungscode verwendet werden kann, um eine Verbindung zu Ihrem Cluster herzustellen.



Using the AWS CLI

Um die Cluster- und Reader-Endpunkte für Ihren Cluster mit Hilfe der AWS CLI zu finden, führen Sie den Befehl `describe-db-clusters` mit diesen Parametern aus.

Parameter

- **`--db-cluster-identifier`**— Fakultativ. Gibt den Cluster an, für den die Endpunkte zurückgegeben werden sollen. Wenn diese Option weggelassen wird, werden Endpunkte für bis zu 100 Ihrer Cluster zurückgegeben.
- **`--query`**—Fakultativ. Gibt die anzuzeigenden Felder an. Dies ist hilfreich, da die Datenmenge, die Sie anzeigen müssen, um die Endpunkte zu finden, reduziert wird. Wenn diese Option weggelassen wird, werden alle Informationen über einen Cluster zurückgegeben.
- **`--region`**—Fakultativ. Verwenden Sie den Parameter `--region`, um die Region anzugeben, für die Sie den Befehl anwenden möchten. Wenn diese Angabe weggelassen wird, wird Ihre Standardregion verwendet.

Example

Das folgende Beispiel liefert die Werte `DBClusterIdentifier`, Endpunkt (Cluster-Endpunkt) und `ReaderEndpoint` für `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
   --region us-east-1 \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
   --region us-east-1 ^
   --db-cluster-identifier sample-cluster ^
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
```

```
  [
     "sample-cluster",
     27017,
     "sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com",
     "sample-cluster.cluster-ro-corlsfccjozr.us-east-1.docdb.amazonaws.com"
  ]
]
```

Nachdem Sie nun den Cluster-Endpunkt haben, können Sie sich mit `mongo` oder `mongodb` mit dem Cluster verbinden. Weitere Informationen finden Sie unter Verbindung mit Endpunkten herstellen.

Suchen nach dem Endpunkt einer Instance

Sie können den Endpunkt für eine Instance mithilfe der Amazon DocumentDB-Konsole oder derAWS CLI.

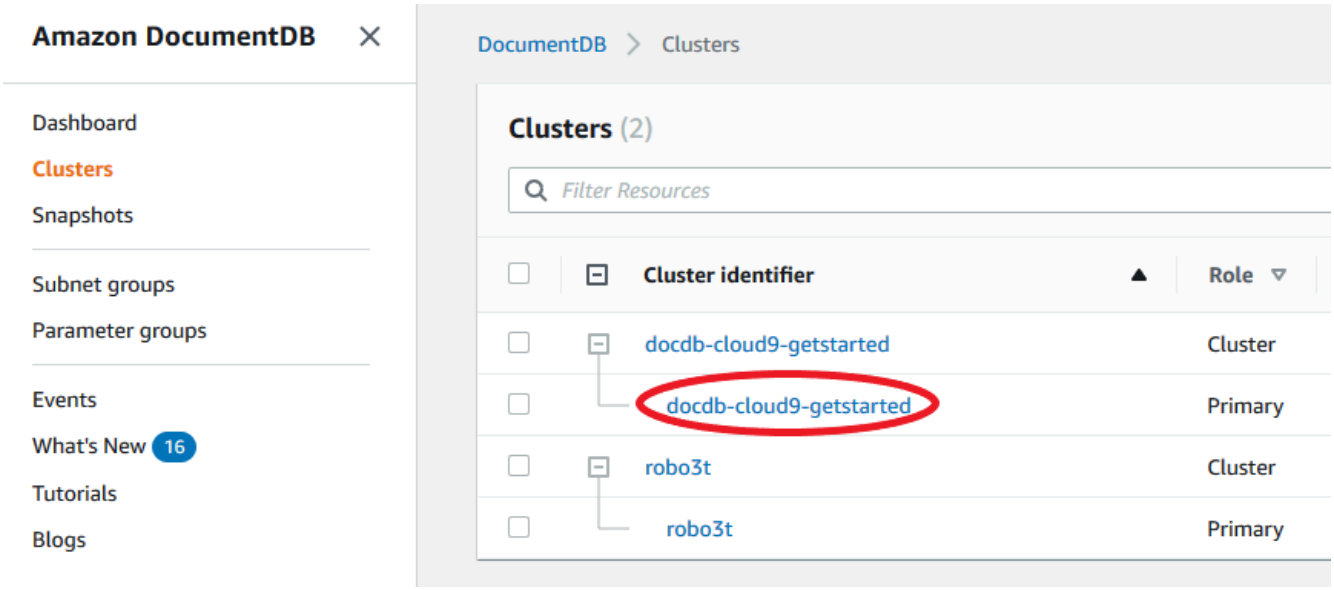
Using the AWS Management Console

So suchen Sie den Endpunkt einer Instance über die Konsole:

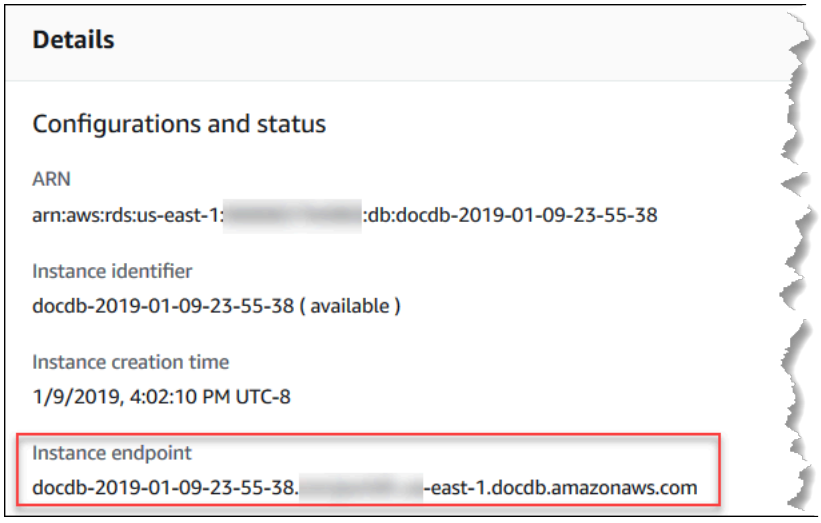
1. Melden Sie sich an beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster).

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Cluster-Navigationsfeld sehen Sie die SpalteCluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Markieren Sie das Kästchen links neben der Instanz, an der Sie interessiert sind.
5. Scrollen Sie nach unten zum Abschnitt Details und suchen Sie dann den Instance-Endpunkt.



6. Um eine Verbindung zu dieser Instance herzustellen, scrollen Sie nach oben zum Abschnitt Connect (Verbinden). Suchen Sie die Verbindungszeichenfolge für die mongo-Shell und eine Verbindungszeichenfolge, die im Anwendungscode verwendet werden kann, um eine Verbindung zu Ihrer Instance herzustellen.

Using the AWS CLI

Um den Instance-Endpunkt mit der AWS CLI zu finden, führen Sie den folgenden Befehl mit diesen Argumenten aus.

Argumente

- **`--db-instance-identifier`**—Fakultativ. Gibt die Instance an, für die der Endpunkt zurückgegeben werden soll. Wenn diese Option weggelassen wird, wird der Endpunkt für bis zu 100 Ihrer Instances zurückgegeben.
- **`--query`**—Fakultativ. Gibt die anzuzeigenden Felder an. Dies ist hilfreich, da die Datenmenge, die Sie anzeigen müssen, um die Endpunkte zu finden, reduziert wird. Wenn diese Option weggelassen wird, werden alle Informationen zu einer Instance zurückgegeben. Das Feld `Endpoint` hat drei Mitglieder, sodass die Auflistung in der Abfrage wie im folgenden Beispiel alle drei Mitglieder zurückgibt. Wenn Sie nur an einigen der `Endpoint`-Mitglieder interessiert sind, ersetzen Sie `Endpoint` in der Abfrage durch die Mitglieder, die Sie interessieren, wie im zweiten Beispiel.
- **`--region`**— Fakultativ. Verwenden Sie den Parameter `--region`, um die Region anzugeben, für die Sie den Befehl anwenden möchten. Wenn diese Angabe weggelassen wird, wird Ihre Standardregion verwendet.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

Für Windows:

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "sample-cluster-instance",
        {
            "Port": 27017,
            "Address": "sample-cluster-instance.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "HostedZoneId": "Z2R2ITUGPM61AM"
        }
    ]
]
```

Wenn Sie die Ausgabe reduzieren, um die `HostedZoneId` des Endpunkts zu eliminieren, können Sie Ihre Abfrage ändern, indem Sie `Endpoint.Port` und `Endpoint.Address` festlegen.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

Für Windows:

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    [
        "sample-cluster-instance",
        27017,
        "sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com"
    ]
]
```

Nachdem Sie nun den Instance-Endpunkt haben, können Sie sich mit mongo oder mongodb mit der Instance verbinden. Weitere Informationen finden Sie unter Verbindung mit Endpunkten herstellen.

Verbindung mit Endpunkten herstellen

Wenn Sie Ihren Endpunkt, entweder Cluster oder Instance, haben, können Sie sich über die mongo-Shell oder eine Verbindungszeichenfolge mit ihm verbinden.

Verbindung mit der Mongo-Shell herstellen

Verwenden Sie die folgende Struktur, um die Zeichenfolge zu erstellen, die Sie für die Verbindung zu Ihrem Cluster oder Ihrer Instance über die mongo-Shell benötigen:

```
mongo \
    --ssl \
    --host Endpoint:Port \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

mongo-Shell-Beispiele

Verbinden mit einem Cluster:

```
mongo \
    --ssl \
    --host sample-cluster.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Verbindung zu einer Instance herstellen:

```
mongo \
    --ssl \
    --host sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
```

```
    --password Password
```

Verbinden über eine Verbindungszeichenfolge

Verwenden Sie die folgende Struktur, um die Verbindungszeichenfolge zu erstellen, die Sie für die Verbindung zu Ihrem Cluster oder Ihrer Instance benötigen.

```
mongodb://UserName:Password@endpoint:port?replicaSet=rs0&ssl_ca_certs=global-
bundle.pem
```

Beispiele für Verbindungszeichenfolgen

Verbinden mit einem Cluster:

```
mongodb://UserName:Password@sample-cluster.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Verbindung zu einer Instance herstellen:

```
mongodb://UserName:Password@sample-cluster-instance.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Grundlegendes zu Amazon DocumentDB-Amazon-Ressourcennamen (ARNs)

Ressourcen, die Sie in AWS erstellen, sind jeweils eindeutig mit einem Amazon-Ressourcennamen (ARN) gekennzeichnet. Für bestimmte Amazon DocumentDB-Operationen (mit MongoDB-Kompatibilität) müssen Sie eine Amazon DocumentDB-Ressource eindeutig identifizieren, indem Sie ihren ARN angeben. Wenn Sie beispielsweise ein Tag zu einer Ressource hinzufügen, müssen Sie den ARN der Ressource angeben.

Themen

Erstellen eines ARN für eine Amazon DocumentDB-Ressource

Sie können einen ARN für eine Amazon DocumentDB-Ressource mit der folgenden Syntax erstellen. Amazon DocumentDB teilt sich das Format von Amazon Relational Database Service (Amazon RDS) ARNS. Amazon DocumentDB-ARNs enthalten `rds` und nicht `docdb`.

`arn:aws:rds:`*`region`*`:`*`account_number`*`:`*`resource_type`*`:`*`resource_id`*

Name der Region	Region	Availability Zones (Datenverarbeitung)
USA Ost (Ohio)	`us-east-2`	3
USA Ost (Nord-Virginia)	`us-east-1`	6
USA West (Oregon)	`us-west-2`	4
Südamerika (São Paulo)	`sa-east-1`	3
Asien-Pazifik (Hongkong)	`ap-east-1`	3
Asien-Pazifik (Hyderabad)	`ap-south-2`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	4
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Kanada (Zentral)	`ca-central-1`	3

Name der Region	Region	Availability Zones (Datenverarbeitung)
Region China (Peking)	`cn-north-1`	3
China (Ningxia)	`cn-northwest-1`	3
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
AWS GovCloud (USA-West)	`us-gov-west-1`	3
AWS GovCloud (USA-Ost)	`us-gov-east-1`	3

> **Note**
>
> Die Amazon DocumentDB-Architektur trennt Speicher und Datenverarbeitung. Für die Speicherebene repliziert Amazon DocumentDB sechs Kopien Ihrer Daten über drei AWS Availability Zones (AZs). Die AZs in der obigen Tabelle sind die Anzahl an AZs, die Sie in einer bestimmten Region für die Bereitstellung von Datenverarbeitungs-Instances verwenden können. Wenn Sie beispielsweise einen Amazon DocumentDB-Cluster in ap-northeast-1 starten, wird Ihr Speicher auf sechs Arten über drei AZs repliziert, Ihre Datenverarbeitungs-Instances sind jedoch nur in zwei AZs verfügbar.

Die folgende Tabelle zeigt das Format, das Sie beim Erstellen eines ARN für eine bestimmte Amazon DocumentDB-Ressource verwenden sollten. Amazon DocumentDB teilt sich das Format von Amazon RDS ARNS. Amazon DocumentDB-ARNs enthalten `rds` und nicht `docdb`.

Ressourcentyp	ARN-Format/Beispiel
Instance (db)	arn:aws:rds: *region*:*account_number* :db:*resource_id* arn:aws:rds:us-east-1: *1234567890* :db:*sample-db-instance*
Cluster (cluster)	arn:aws:rds: *region*:*account_number* :cluster: *resource_id* arn:aws:rds:us-east-1: *1234567890* :cluster: *sample-db-cluster*
Cluster-Parametergruppe (cluster-pg)	arn:aws:rds: *region*:*account_number* :cluster-pg: *resource_id* arn:aws:rds:us-east-1: *1234567890* :cluster-pg: *sample-db-cluster-parameter-group*
Sicherheitsgruppe (secgrp)	arn:aws:rds: *region*:*account_number* :secgrp:*resource_id* arn:aws:rds:us-east-1: *1234567890* :secgrp:*sample-public-secgrp*
Cluster-Snapshot (cluster-snapshot)	arn:aws:rds: *region*:*account_number* :cluster-snapshot: *resource_id* arn:aws:rds:us-east-1: *1234567890* :cluster-snapshot: *sample-db-cluster-snapshot*
Subnetzgruppe (subgrp)	arn:aws:rds: *region*:*account_number* :subgrp:*resource_id*

Ressourcentyp	ARN-Format/Beispiel
	`arn:aws:rds:us-east-1:` *`1234567890`* `:subgrp:`*`sample-subnet-10`*

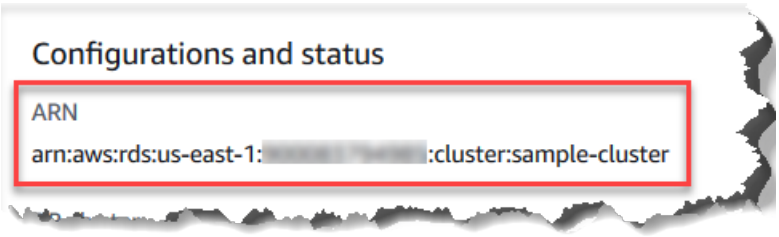
Suchen eines Amazon DocumentDB-Ressourcen-ARN

Sie finden den ARN einer Amazon DocumentDB-Ressource mithilfe der AWS Management Console oder der AWS CLI.

Using the AWS Management Console

Um einen ARN über die Konsole zu suchen, navigieren Sie zu der Ressource, für die Sie einen ARN erhalten möchten, und zeigen Sie die Details für diese Ressource an.

Beispielsweise können Sie den ARN für einen Cluster im Bereich Details für den Cluster erhalten, wie im folgenden Screenshot gezeigt.



Using the AWS CLI

Um einen ARN mit der AWS CLI für eine bestimmte Amazon DocumentDB-Ressource abzurufen, verwenden Sie die -`describe`Operation für diese Ressource. Die folgende Tabelle zeigt jede AWS CLI-Operation und die ARN-Eigenschaft, die mit der Operation verwendet wird, um einen ARN zu erhalten.

AWS CLI-Befehl	ARN-Eigenschaft
`describe-db-instances`	`DBInstanceArn`
`describe-db-clusters`	`DBClusterArn`
`describe-db-parameter-groups`	`DBParameterGroupArn`

AWS CLI-Befehl	ARN-Eigenschaft
`describe-db-cluster-parameter-groups`	`DBClusterParameterGroupArn`
`describe-db-security-groups`	`DBSecurityGroupArn`
`describe-db-snapshots`	`DBSnapshotArn`
`describe-db-cluster-snapshots`	`DBClusterSnapshotArn`
`describe-db-subnet-groups`	`DBSubnetGroupArn`

Example - ARN Ihres Clusters suchen

Die folgende AWS CLI-Operation findet den ARN für den Cluster `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster"
]
```

Example - ARNs für mehrere Parametergruppen suchen

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
```

```
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:custom3-6-param-grp",
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora5.6",
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.docdb3.6"
]
```

Taggen von Amazon DocumentDB-Ressourcen

Sie können Amazon DocumentDB-Tags (mit MongoDB-Kompatibilität) verwenden, um Ihren Amazon DocumentDB-Ressourcen Metadaten hinzuzufügen. Diese Tags können verwendet werden mitAWS Identity and Access Management(IAM) -Richtlinien zur Verwaltung des Zugriffs auf Amazon DocumentDB-Ressourcen und zur Steuerung, welche Aktionen auf die Ressourcen angewendet werden können. Sie können Tags auch verwenden, um Kosten zu verfolgen, indem Ausgaben für ähnlich markierte Ressourcen gruppiert werden.

Sie können die folgenden Amazon DocumentDB-Ressourcen taggen:

- Cluster
- Instances
- Snapshots
- Cluster-Snapshots
- Parametergruppen
- Cluster-Parametergruppen
- Sicherheitsgruppen
- Subnetzgruppen

Überblick über Amazon DocumentDB-Ressourcen-Tags

Ein Amazon DocumentDB-Tag ist ein Name-Wert-Paar, das Sie definieren und mit einer Amazon DocumentDB-Ressource verknüpfen. Der Name wird als der Schlüssel bezeichnet. Die Angabe eines Wertes für den Schlüssel ist optional. Sie können Tags verwenden, um einer Amazon DocumentDB-Ressource beliebige Informationen zuzuweisen. Sie können einen Tag-Schlüssel z. B. dazu verwenden, um eine Kategorie zu definieren, und ein Tag-Wert könnte ein Element in dieser Kategorie sein. Sie könnten beispielsweise einen Tag-Schlüssel von definieren`project`und einen Tag-Wert von`Salix`, was darauf hinweist, dass die Amazon DocumentDB-Ressource dem Salix-Projekt zugewiesen ist. Sie können Tags auch verwenden, um Amazon DocumentDB-Ressourcen so zu kennzeichnen, dass sie für Test- oder Produktionszwecke verwendet werden, indem Sie einen Schlüssel wie`environment=test`oder`environment=production`. Wir empfehlen Ihnen, einen konsistenten Satz von Tag-Schlüsseln zu verwenden, um die Nachverfolgung von Metadaten zu erleichtern, die Amazon DocumentDB-Ressourcen zugeordnet sind.

Sie können Tags (Markierungen) auch zum Organisieren Ihrer AWS-Kontorechnung verwenden, um Ihre eigene Kostenstruktur darzustellen. Dazu müssen Sie sich registrieren, um Ihre AWS-Konto-Rechnung mit Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können. Weitere Informationen finden Sie unterVerwenden von Tags für die Kostenzuweisungin derAWSBenutzerhandbuch für Abrechnung und Kostenmanagement.

Jede Amazon DocumentDB-Ressource hat einen Tagsatz, der alle Tags enthält, die dieser Ressource zugewiesen sind. Ein Tag-Satz kann bis zu zehn Tags enthalten oder leer sein. Wenn Sie einer Amazon DocumentDB-Ressource ein Tag hinzufügen, das denselben Schlüssel wie ein vorhandenes Tag auf der Ressource hat, überschreibt der neue Wert den alten Wert.

AWS wendet auf Ihre Tags keine semantische Bedeutung an. Tags werden als reine Zeichenfolgen interpretiert. Amazon DocumentDB kann je nach den Einstellungen, die Sie bei der Erstellung der Ressource verwenden, Tags für eine Instance oder andere Amazon DocumentDB-Ressourcen festlegen. Amazon DocumentDB könnte beispielsweise ein Tag hinzufügen, das angibt, dass eine Instance für die Produktion oder zum Testen bestimmt ist.

Sie können ein Tag an einen Snapshot anfügen. Diese Gruppierung taucht jedoch nicht in Ihrer Rechnung auf.

Sie können das verwendenAWS Management Consoleoder dasAWS CLIum Tags zu Amazon DocumentDB-Ressourcen hinzuzufügen, aufzulisten und zu löschen. Bei der AWS CLI müssen Sie den Amazon-Ressourcennamen (ARN) für die zu verwendende Ressource angeben. Weitere Informationen zu Amazon DocumentDB-ARNs finden Sie unterGrundlegendes zu Amazon DocumentDB-Amazon-Ressourcennamen (ARNs).

Tag-Einschränkungen

Die folgenden Einschränkungen gelten für Amazon DocumentDB-Tags:

- Maximale Anzahl von Tags pro Ressource: 10
- Maximale Schlüssellänge – 128 Unicode-Zeichen
- Maximale Wertlänge – 256 Unicode-Zeichen
- Gültige Zeichen für Schlüssel und Wert – Groß- und Kleinbuchstaben im UTF-8-Zeichensatz, Ziffern, Leerzeichen und die folgenden Zeichen: `_ . : / = + -` und `@` (Java-Regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`)
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Das Präfix `aws:` kann nicht für die Tag-Schlüssel oder Werte verwendet werden. Dieses Präfix ist für AWS reserviert.

Hinzufügen und Aktualisieren von Tags auf einer Amazon DocumentDB-Ressource

Sie können einer Ressource bis zu 10 Tags hinzufügen, indem Sie denAWS Management Consoleoder dasAWS CLI.

Using the AWS Management Console

Der Prozess für das Hinzufügen eines Tags zu einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie das Tag hinzufügen. In diesem Beispiel fügen Sie ein Tag zu einem Cluster hinzu.

So fügen Sie Tags einem Cluster über die Konsole hinzu oder aktualisieren sie

1. Melden Sie sich an beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus.

3. Wählen Sie den Namen des Clusters aus, zu dem Sie Tags hinzufügen möchten.
4. Scrollen Sie nach unten zum Abschnitt Tags und wählen Sie anschließend Bearbeiten aus.
5. Für alle Tags, die Sie dieser Ressource hinzufügen möchten, gehen Sie wie folgt vor:

    a. Um ein neues Tag hinzuzufügen, geben Sie den Namen des Tags in das Feld Schlüssel ein. Um den Wert eines Tags zu ändern, suchen Sie den Namen des Tags in der Spalte Schlüssel.

    b. Um dem Tag einen neuen oder aktualisierten Wert zu geben, geben Sie einen Wert für das Tag in das Feld Wert ein.

    c. Wenn Sie mehrere Tags hinzufügen möchten, klicken Sie auf Hinzufügen. Andernfalls klicken Sie auf Speichern, sobald Sie fertig sind.

Using the AWS CLI

Der Prozess für das Hinzufügen eines Tags zu einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie die Tags hinzufügen. In diesem Beispiel fügen Sie drei Tags zu einem Cluster hinzu. Das zweite Tag, `key2`, hat keinen Wert.

Verwenden Sie die AWS CLI-Operation `add-tags-to-resource` mit diesen Parametern.

Parameter

- **`--resource-name`**— Der ARN der Amazon DocumentDB-Ressource, der Sie Tags hinzufügen möchten.
- **`--tags`**—Eine Liste der Tags (Schlüssel-Wert-Paar), die Sie dieser Ressource hinzufügen möchten, im folgenden Format`Key=`*`key-name`*`,Value=`*`tag-value`*.

Example

Für Linux, macOS oder Unix:

```
aws docdb add-tags-to-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

Für Windows:

```
aws docdb add-tags-to-resource ^
```

```
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

Die `add-tags-to-resource`-Operation erzeugt keine Ausgabe. Um die Ergebnisse der Operation anzuzeigen, verwenden Sie die `list-tags-for-resource`-Operation.

Tags auf einer Amazon DocumentDB-Ressource auflisten

Sie können das verwendenAWS Management Consoleoder dasAWS CLIum eine Liste der Tags für eine Amazon DocumentDB-Ressource zu erhalten.

Using the AWS Management Console

Der Prozess für das Auflisten von Tags in einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie das Tag hinzufügen. In diesem Beispiel listen Sie die Tags für einen Cluster auf.

So listen Sie die Tags in einem Cluster mithilfe der Konsole auf

1. Öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie Tags auflisten möchten.
4. Um eine Liste der Tags in dieser Ressource anzuzeigen, scrollen Sie nach unten zum Abschnitt Tags.

Using the AWS CLI

Der Prozess für das Auflisten der Tags in einer Ressource ist ähnlich, unabhängig davon, für welche Ressource Sie das Tag auflisten. In diesem Beispiel listen Sie die Tags in einem Cluster auf.

Verwenden Sie die AWS CLI-Operation `list-tags-for-resource` mit diesen Parametern.

Parameter

- **`--resource-name`**—Erforderlich. Der ARN der Amazon DocumentDB-Ressource, für die Sie Tags auflisten möchten.

Example

Für Linux, macOS oder Unix:

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

Für Windows:

```
aws docdb list-tags-for-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": ""
        },
        {
            "Key": "key3",
            "Value": "value3"
        }
    ]
}
```

Tags aus einer Amazon DocumentDB-Ressource entfernen

Sie können das verwendenAWS Management Consoleoder dasAWS CLIum Tags aus Amazon DocumentDB-Ressourcen zu entfernen.

Using the AWS Management Console

Der Prozess für das Entfernen von Tags aus einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie das Tag hinzufügen. In diesem Beispiel entfernen Sie Tags aus einem Cluster.

So entfernen Sie Tags aus einem Cluster mithilfe der Konsole

1. Öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus.
3. Wählen Sie den Namen des Clusters aus, aus dem Sie Tags entfernen möchten.
4. Scrollen Sie nach unten zum Abschnitt Tags und wählen Sie anschließend Bearbeiten aus.
5. Wenn Sie alle Tags aus dieser Ressource entfernen möchten, wählen Sie Remove all (Alle entfernen) aus. Andernfalls gehen Sie für alle Tags, die Sie aus dieser Ressource entfernen möchten, wie folgt vor:

   a. Suchen Sie den Namen des Tags in der Spalte Schlüssel.

   b. Wählen Sie Entfernen auf derselben Zeile wie der Tag-Schlüssel aus.

   c. Wenn Sie fertig sind, wählen Sie Speichern aus.

Using the AWS CLI

Der Prozess für das Entfernen eines Tags aus einer Ressource ist ähnlich, unabhängig davon, aus welcher Ressource Sie das Tag entfernen. In diesem Beispiel entfernen Sie ein Tag aus einem Cluster.

Verwenden Sie die AWS CLI-Operation `remove-tags-from-resource` mit diesen Parametern.

- **`--resource-name`**—Erforderlich. Der ARN der Amazon DocumentDB-Ressource, aus der Sie Tags entfernen möchten.
- **`--tag-keys`**—Erforderlich. Eine Liste der Tag-Schlüssel, die von dieser Ressource entfernt werden sollen.

Example

Für Linux, macOS oder Unix:

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Für Windows:

```
aws docdb remove-tags-from-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Die `removed-tags-from-resource`-Operation erzeugt keine Ausgabe. Um die Ergebnisse der Operation anzuzeigen, verwenden Sie die `list-tags-for-resource`-Operation.

Verwalten von Amazon DocumentDB

In regelmäßigen Abständen führt Amazon DocumentDB Wartungsarbeiten an Amazon DocumentDB-Ressourcen durch. Diese Wartung umfasst in den meisten Fällen Aktualisierungen der Datenbank-Engine (Cluster-Wartung) oder des zugrunde liegenden Betriebssystems (OS) der Instance (Instance-Wartung). Datenbank-Engine-Updates sind erforderliche Patches und enthalten Sicherheitskorrekturen, Fehlerbehebungen und Verbesserungen der Datenbank-Engine. Updates des Betriebssystems enthalten häufig Sicherheitskorrekturen. Betriebssystem-Patches sind zwar optional, wir empfehlen jedoch, sie auf Ihre Amazon DocumentDB-Instances anzuwenden, sobald sie verfügbar sind.

Datenbank-Engine-Patches erfordern, dass Sie Ihre Amazon DocumentDB-Cluster für kurze Zeit offline schalten. Sobald diese Patches verfügbar sind, werden sie automatisch für die Anwendung während eines bevorstehenden geplanten Wartungsfensters Ihres Amazon DocumentDB-Clusters geplant.

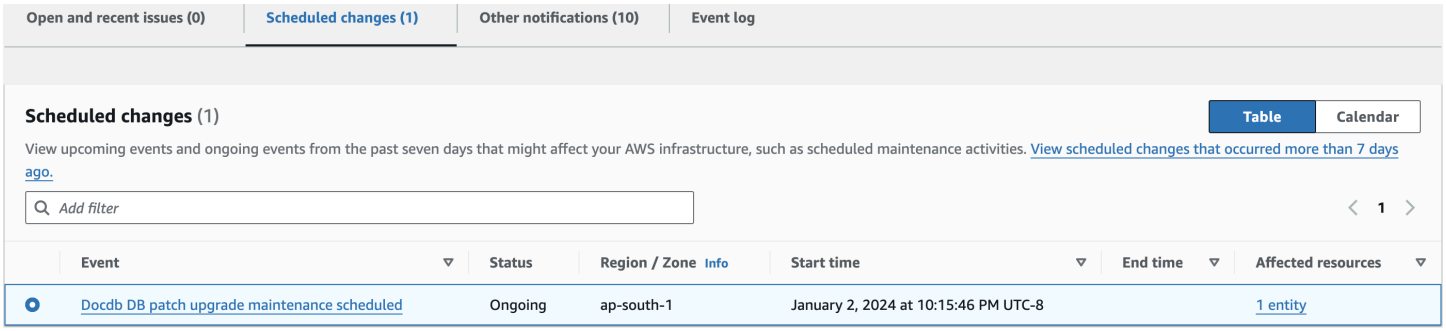
Sowohl die Cluster- als auch die Instance-Wartung haben jeweils eigene Wartungszeitfenster. Cluster- und Instance-Änderungen, die Sie nicht sofort anwenden möchten, werden auch während des Wartungsfensters angewendet. Wenn Sie einen Cluster erstellen, weist Amazon DocumentDB standardmäßig sowohl für einen Cluster als auch für jede einzelne Instance ein Wartungsfenster zu. Beim Anlegen eines Clusters oder einer Instance können Sie das Wartungsfenster auswählen. Sie können die Wartungsfenster außerdem jederzeit an Ihre Geschäftspläne oder -praktiken anpassen. Es wird generell empfohlen, Wartungsfenster auszuwählen, die die Auswirkungen der Wartung auf Ihre Anwendung minimieren (z. B. an Abenden oder Wochenenden). Diese Leitlinie ist stark kontextabhängig. Sie variiert je nach Art der Anwendung und der Nutzungsmuster.

Themen

- Vom Benutzer initiierte Updates
- Verwalten Ihrer Amazon DocumentDB-Wartungsfenster
- Arbeiten mit Betriebssystem-Updates

Benachrichtigungen für Amazon DocumentDB-Engine-Patches

Sie erhalten Wartungsbenachrichtigungen für erforderliche Datenbank-Engine-Patches über Zustandsereignisse in der AWS Health Dashboard (AHD) in der -AWSKonsole und über E-Mails. Wenn ein Wartungspatch der Amazon DocumentDB-Engine in einer bestimmten AWS Region verfügbar wird, erhalten alle betroffenen Amazon DocumentDB-Benutzerkonten in der Region eine AHD- und E-Mail-Benachrichtigung für jede vom Patch betroffene Amazon DocumentDB-Version. Sie können diese Benachrichtigungen im Abschnitt Geplante Änderungen der AHD in der -AWSKonsole anzeigen. Die Benachrichtigung enthält Details zum Zeitpunkt der Patch-Verfügbarkeit, den Zeitplan für die automatische Anwendung, die Liste der betroffenen Cluster und Versionshinweise. Diese Benachrichtigung wird auch per E-Mail an die E-Mail-Adresse des Stammbenutzers des AWS Kontos zugestellt.



Sobald Sie diese Benachrichtigung erhalten haben, können Sie diese Engine-Patches vor dem geplanten Datum der automatischen Anwendung selbst auf Ihre Amazon DocumentDB-Cluster anwenden. Oder Sie können warten, bis die Engine-Patches während eines bevorstehenden Wartungsfensters automatisch angewendet werden (Standardoption).

> **Note**
>
> Der Status für die Benachrichtigung in der AHD wird auf „Laufend“ gesetzt, bis ein neuer Amazon DocumentDB-Engine-Patch mit einer neuen Engine-Patch-Version veröffentlicht wird.
> Sobald der Engine-Patch auf Ihren Amazon DocumentDB-Cluster angewendet wurde, wird die Engine-Patch-Version des Clusters aktualisiert, um die Version in der Benachrichtigung

widerzuspiegeln. Sie können den `db.runCommand({getEngineVersion: 1})` Befehl ausführen, um dieses Update zu überprüfen.

AWS Health ist auch in Amazon integriert EventBridge , das Ereignisse verwendet, um skalierbare ereignisgesteuerte Anwendungen zu erstellen, und in über 20 Ziele integriert werden kann, darunter AWS Lambda, Amazon Simple Queue Service (SQS) und andere. Sie können `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_SCHEDULED` Ereigniscode verwenden, um Amazon einzurichten, EventBridge bevor Engine-Patches verfügbar sind. Sie können so einrichten EventBridge , dass es auf das Ereignis reagiert und automatisch Aktionen ausführt, z. B. das Erfassen von Ereignisinformationen, das Initiieren zusätzlicher Ereignisse, das Senden von Benachrichtigungen über zusätzliche Kanäle wie Push-Benachrichtigungen an die AWS Console Mobile Applicationund das Ergreifen korrigierender oder anderer Aktionen, wenn ein Amazon DocumentDB-Engine-Patch in Ihrer Region verfügbar wird.

Im seltenen Szenario, dass Amazon DocumentDB einen Engine-Patch bricht, erhalten Sie eine AHD-Benachrichtigung sowie eine E-Mail, die Sie über den Abbruch informiert. Dementsprechend können Sie den `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_CANCELLED` Ereigniscode verwenden, um Amazon so einzurichten EventBridge , dass es auf dieses Ereignis reagiert. Weitere Informationen zur Verwendung von Amazon- EventBridge Regeln finden Sie im Amazon-EventBridge Benutzerhandbuch.

Anzeigen ausstehender Amazon DocumentDB-Wartungsaktionen

Sie können mit der AWS Management Console oder AWS CLI sehen, ob für Ihren Cluster ein Wartungsupdate verfügbar ist.

Wenn ein Update verfügbar ist, können Sie eine der folgenden Aktionen durchführen:

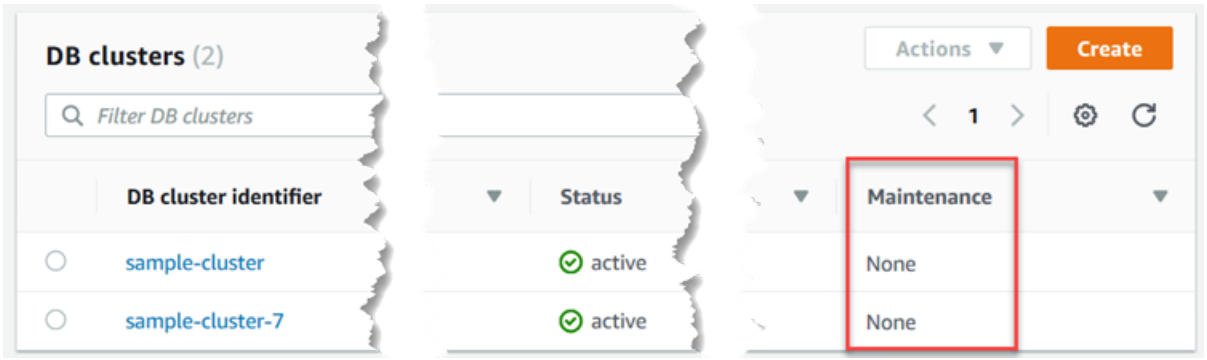
- Verschieben Sie eine Wartungsaktion, die derzeit für das nächste Wartungsfenster geplant ist (nur für Betriebssystem-Patches).
- Sofortiges Durchführen der Wartungsaktivitäten-
- Einplanen der Wartungsaktivitäten für das nächste Wartungsfenster.

> **Note**
>
> Wenn Sie keine Maßnahmen ergreifen, werden die erforderlichen Wartungsaktionen wie Engine-Patches automatisch in einem bevorstehenden geplanten Wartungsfenster angewendet.

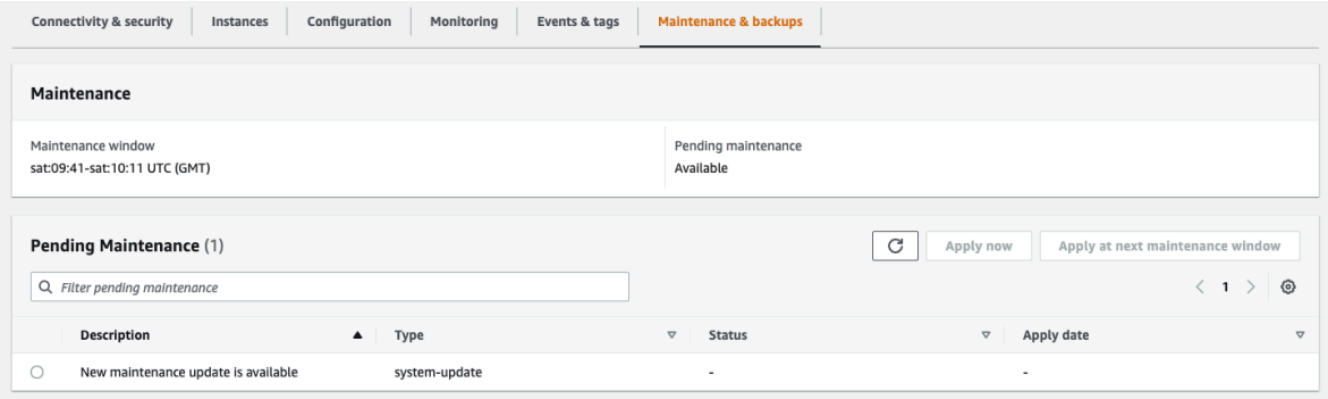
Das Wartungsfenster legt fest, wann die ausstehenden Operationen gestartet werden, gibt aber kein Abschlussdatum für diese Operationen vor; die Ausführungsdauer ist weder bekannt noch a priori beschränkt.

Using the AWS Management Console

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wenn ein Update verfügbar ist, wird es in der Spalte Wartung für den Cluster in der Amazon DocumentDB-Konsole durch das Wort Verfügbares , Erforderlich oder Nächstes Fenster angezeigt, wie hier gezeigt:



4. Um eine Aktion auszuführen, wählen Sie den Cluster aus, um seine Details anzuzeigen, und wählen Sie dann Wartung und Backups aus. Die Elemente Ausstehende Wartung werden angezeigt.



Using the AWS CLI

Verwenden Sie die folgende AWS CLI-Operation, um festzustellen, welche Wartungsarbeiten noch ausstehen. Die Ausgabe hier zeigt keine offenen Wartungsaktionen.

```
aws docdb describe-pending-maintenance-actions
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "PendingMaintenanceActions": []
}
```

Anwenden von Amazon DocumentDB-Engine-Updates

Mit Amazon DocumentDB können Sie wählen, wann Wartungsvorgänge angewendet werden sollen. Sie können entscheiden, wann Amazon DocumentDB Updates anwendet, indem Sie die AWS Management Console oder verwendenAWS CLI.

Verwenden Sie die Verfahren in diesem Thema, um Ihr Cluster sofort zu aktualisieren oder ein Upgrade zu planen.

Using the AWS Management Console

Sie können die Konsole verwenden, um Updates für Ihre Amazon DocumentDB-Cluster zu verwalten.

So verwalten Sie ein Update für einen Cluster

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie in der Liste der Cluster die Schaltfläche neben dem Namen des Clusters, für den Sie den Wartungsvorgang anwenden möchten.
4. Wählen Sie aus dem Menü Actions (Aktionen) eine der folgenden Optionen aus:

   - Upgrade now (Upgrade jetzt), um die anstehenden Wartungsarbeiten sofort durchzuführen.
   - Upgrade at next window (Upgrade im nächsten Fenster), um die anstehenden Wartungsarbeiten im nächsten Wartungsfenster des Clusters durchzuführen.

   Alternativ können Sie auf der Registerkarte Wartung und Backups des Clusters im Abschnitt Ausstehende Wartung auf Jetzt anwenden oder Beim nächsten Wartungsfenster anwenden klicken (siehe Verwenden von AWS Management Console ist der vorherige Abschnitt).

   > **Note**
   >
   > Wenn keine Wartungsaufgaben ausstehen, sind alle oben genannten Optionen inaktiv.

Using the AWS CLI

Um ein ausstehendes Update auf einen Cluster anzuwenden, verwenden Sie die `apply-pending-maintenance-action`AWS CLIOperation.

Parameter

- **`--resource-identifier`**– Der Amazon-Ressourcenname (ARN) von Amazon DocumentDB der Ressource, für die die ausstehende Wartungsaktion gilt.
- **`--apply-action`**– Die ausstehende Wartungsaktion, die auf diese Ressource angewendet werden soll.

  Gültige Werte: `system-update` und `db-upgrade`.

- **`--opt-in-type`**– Ein Wert, der den Typ der Opt-In-Anforderung angibt oder eine Opt-In-Anforderung rückgängig macht. Eine Opt-in-Anfrage vom Typ `immediate` kann nicht rückgängig gemacht werden.

  Zulässige Werte:

  - `immediate`– Wenden Sie die Wartungsaktion sofort an.
  - `next-maintenance`– Wenden Sie die Wartungsaktion während des nächsten Wartungsfensters für die Ressource an.
  - `undo-opt-in`– Bricht alle vorhandenen `next-maintenance` Opt-In-Anforderungen ab.

Example

Für Linux, macOS oder Unix:

```
aws docdb apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb \
    --apply-action system-update \
    --opt-in-type immediate
```

Für Windows:

```
aws docdb apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb ^
    --apply-action system-update ^
    --opt-in-type immediate
```

Um eine Liste von Ressourcen zurückzugeben, für die mindestens ein Update ansteht, verwenden Sie die AWS CLI-Operation `describe-pending-maintenance-actions`.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

Für Windows:

```
aws docdb describe-pending-maintenance-actions ^
```

```
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:001234567890:cluster:sample-cluster",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "system-update",
                    "CurrentApplyDate": "2019-01-11T03:01:00Z",
                    "Description": "db-version-upgrade",
                    "ForcedApplyDate": "2019-01-18T03:01:00Z",
                    "AutoAppliedAfterDate": "2019-01-11T03:01:00Z"
                }
            ]
        }
    ]
}
```

Sie können auch eine Liste von Ressourcen für einen Cluster zurückgeben, indem Sie den `--filters` Parameter der`describe-pending-maintenance-actions` AWS CLI Operation angeben. Das Format für die `--filters`-Operation ist `Name=`*`filter-name`*`,Values=`*`resource-id`*`,...`.

`db-cluster-id` ist die zulässigen Werte für den `Name` Parameter des Filters. Dieser Wert akzeptiert eine Liste von Cluster-IDs oder ARNs . In der zurückgegebenen Liste sind nur die aussehenden Wartungsaktionen für die Cluster aufgeführt, die diesen IDs bzw. ARNs entsprechen.

Das folgende Beispiel gibt die anstehenden Wartungsaktionen für die Cluster `sample-cluster1` und `sample-cluster2` zurück.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-actions \
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Für Windows:

```
aws docdb describe-pending-maintenance-actions ^
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Anwenden von Daten

Jede Wartungsaktivität hat ein entsprechendes Anwendungsdatum, das Sie bei der Beschreibung der anstehenden Wartungsaktivität finden. Die Ausgabe der ausstehenden Wartungsaktivitäten aus der AWS CLI führt drei Termine auf:

- **`CurrentApplyDate`**– Das Datum, an dem die Wartungsaktion entweder sofort oder während des nächsten Wartungsfensters angewendet wird. Wenn die Wartung optional ist, kann dieser Wert `null` sein.
- **`ForcedApplyDate`**– Das Datum, an dem die Wartung unabhängig von Ihrem Wartungsfenster automatisch angewendet wird.
- **`AutoAppliedAfterDate`**– Das Datum, nach dem die Wartung während des Wartungsfensters des Clusters angewendet wird.

Vom Benutzer initiierte Updates

Als Amazon DocumentDB-Benutzer können Sie Aktualisierungen Ihrer Cluster oder Instances initiieren. Sie können beispielsweise die Klasse einer Instance in eine Klasse mit mehr oder weniger Speicher ändern oder die Parametergruppe eines Clusters ändern. Amazon DocumentDB zeigt diese Änderungen anders an als von Amazon DocumentDB initiierte Updates. Weitere Informationen zum Ändern eines Clusters oder einer Instance finden Sie in den folgenden Artikeln:

- Ändern eines Amazon DocumentDB-Clusters
- Ändern einer Amazon DocumentDB-Instance

Um eine Liste der ausstehenden, vom Benutzer initiierten Änderungen anzuzeigen, führen Sie den folgenden Befehl aus.

Example

So zeigen Sie ausstehende, von Benutzern initiierte Änderungen für Ihre Instances an:

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

Für Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

In diesem Fall hat `sample-cluster-instance` eine ausstehende Änderung für eine `db.r5.xlarge` Instance-Klasse, während `sample-cluster-instance-2` keine ausstehenden Änderungen hat.

```
[
    [
        "sample-cluster",
        "sample-cluster-instance",
        {
            "DBInstanceClass": "db.r5.xlarge"
        }
    ],
    [
        "sample-cluster",
        "sample-cluster-instance-2",
        {}
    ]
]
```

Verwalten Ihrer Amazon DocumentDB-Wartungsfenster

Jede Instance und jeder Cluster hat ein wöchentliches Wartungsfenster, in dem alle anstehenden Änderungen übernommen werden. Über das Wartungsfenster können Sie kontrollieren, wann angeforderte und erforderliche Änderungen und Software-Patches erfolgen. Wenn ein Wartungsereignis für eine bestimmte Woche geplant ist, wird es während des 30-minütigen Wartungsfensters eingeleitet, das Sie festlegen. Die meisten Wartungsereignisse werden auch

während des 30-minütigen Wartungsfensters abgeschlossen, obwohl größere Wartungsereignisse länger als 30 Minuten dauern können.

Das 30-minütige Wartungsfenster wird zufällig aus einem 8-Stunden-Zeitraum pro Region ausgewählt. Wenn Sie beim Erstellen der Instance oder des Clusters kein bevorzugtes Wartungsfenster angeben, weist Amazon DocumentDB an einem zufällig ausgewählten Wochentag ein 30-minütiges Wartungsfenster zu.

Die folgende Tabelle listet die Blöcke für jede Region auf, von denen Standard-Wartungsfenster zugewiesen werden.

Name der Region	Region	UTC-Zeitblock
USA Ost (Ohio)	us-east-2	03:00-11:00
USA Ost (Nord-Virginia)	us-east-1	03:00-11:00
USA West (Oregon)	us-west-2	06:00-14:00
Asien-Pazifik (Hongkong)	ap-east-1	06:00-14:00
Asien-Pazifik (Hyderabad)	ap-south-2	06:30–14:30
Asien-Pazifik (Mumbai)	ap-south-1	06:00-14:00
Asien-Pazifik (Seoul)	ap-northeast-2	13:00-21:00
Asien-Pazifik (Singapur)	ap-southeast-1	14:00–22:00 Uhr
Asien-Pazifik (Sydney)	ap-southeast-2	12:00–20:00
Asien-Pazifik (Tokio)	ap-northeast-1	13:00-21:00
Kanada (Zentral)	ca-central-1	03:00-11:00
China (Beijing)	cn-north-1	06:00-14:00
China (Ningxia)	cn-northwest-1	06:00-14:00
Europa (Frankfurt)	eu-central-1	21:00–05:00

Name der Region	Region	UTC-Zeitblock
Europa (Irland)	eu-west-1	22:00-06:00
Europa (London)	eu-west-2	22:00-06:00
Europa (Mailand)	eu-south-1	02:00–10:00
Europa (Paris)	eu-west-3	23:59–07:29
South America (São Paulo)	sa-east-1	00:00–08:00
AWS GovCloud (USA-Ost)	us-gov-east-1	17:00–01:00
AWS GovCloud (USA-West)	us-gov-west-1	06:00-14:00

Ändern Ihrer Amazon DocumentDB-Wartungsfenster

Das Wartungsfenster sollte in den Zeitraum mit der geringsten Nutzung fallen und daher unter Umständen von Zeit zu Zeit geändert werden. Ihr Cluster oder Ihre Instance ist während dieser Zeit nur dann nicht verfügbar, wenn Systemänderungen (z. B. eine Speicherskalierung oder ein Instance-Klassen-Wechsel) durchgeführt werden und einen Ausfall erforderlich machen. Die fehlende Verfügbarkeit gilt dann nur für die minimale Zeitspanne, die für die notwendigen Änderungen benötigt wird.

Für Upgrades auf die Datenbank-Engine verwendet Amazon DocumentDB das bevorzugte Wartungsfenster des Clusters und nicht das Wartungsfenster für einzelne Instances.

So ändern Sie das Wartungsfenster

- Für einen Cluster: Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.
- Für eine Instance: Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB-Instance.

Arbeiten mit Betriebssystem-Updates

Instances in Amazon DocumentDB-Clustern erfordern gelegentlich Betriebssystemaktualisierungen. Amazon DocumentDB aktualisiert das Betriebssystem auf eine neuere Version, um die

Datenbankleistung und den allgemeinen Sicherheitsstatus der Kunden zu verbessern. Betriebssystem-Updates ändern weder die Cluster-Engine-Version noch die Instance-Klasse einer Amazon DocumentDB-Instance.

Wir empfehlen, zuerst die Reader-Instances in einem Cluster und dann die Writer-Instance zu aktualisieren, um die Verfügbarkeit Ihres Clusters zu maximieren. Wir empfehlen nicht, Reader- und Writer-Instances gleichzeitig zu aktualisieren, da es im Falle eines Failovers zu längeren Ausfallzeiten kommen kann.

Betriebssystemupdates haben kein Anwendungsdatum und können jederzeit angewendet werden. Wir empfehlen Ihnen, sie regelmäßig anzuwenden, um Ihre Amazon DocumentDB-Datenbanken auf dem neuesten Stand zu halten. Amazon DocumentDB wendet diese Updates nicht automatisch an. Wenn Sie benachrichtigt werden möchten, sobald eine neue optionale Aktualisierung verfügbar ist, können Sie RDS-EVENT-0230 in der Kategorie Sicherheitspatch-Ereignis abonnieren. Informationen zum Abonnieren von Amazon DocumentDB-Ereignissen finden Sie unter Abonnieren von Amazon DocumentDB-Ereignisabonnements.

Sie sollten davon ausgehen, dass, wenn eine Wartung auf Ihrem Cluster oder Ihrer Instance ausgeführt wird, und es sich um eine primäre Instance handelt, ein Failover ausgeführt wird. Um Ihre Verfügbarkeit zu verbessern, empfehlen wir Ihnen, mehr als eine Instance für Ihre Amazon DocumentDB-Cluster zu verwenden. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Failover.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.

> **Important**
>
> Ihre Amazon DocumentDB-Instance wird während des Betriebssystem-Upgrades offline geschaltet.

> **Note**
>
> Es ist möglicherweise erforderlich, im Hinblick auf alle optionalen und obligatorischen Updates auf dem Laufenden zu bleiben, um verschiedene Compliance-Auflagen zu erfüllen. Wir empfehlen Ihnen, alle von Amazon DocumentDB routinemäßig während Ihrer Wartungsfenster zur Verfügung gestellten Updates anzuwenden.

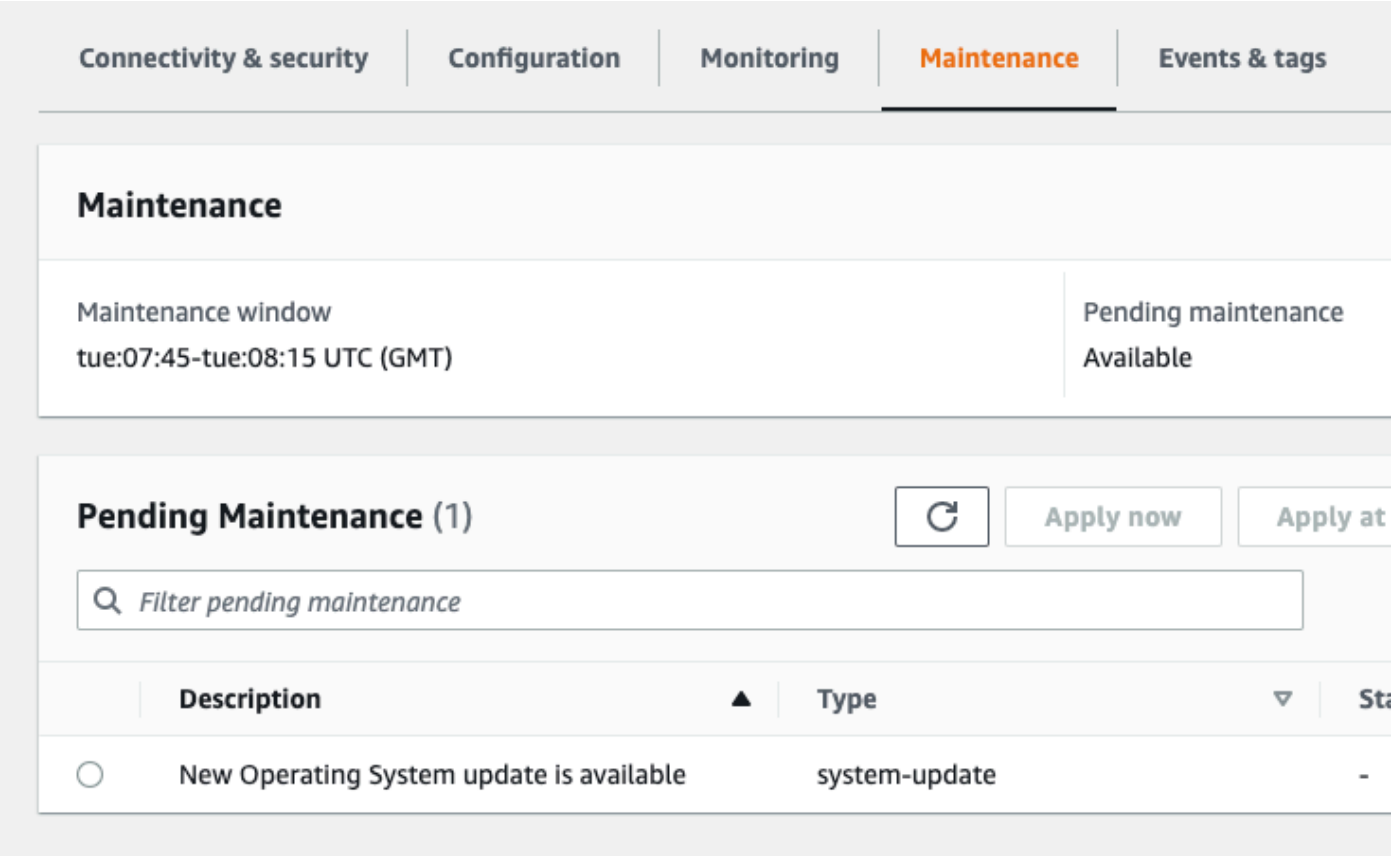
Sie können die AWS Management Console oder die AWS CLI verwenden, um festzustellen, ob ein Update optional oder obligatorisch ist.

Using the AWS Management Console

So ermitteln Sie mithilfe der , ob ein Update optional oder obligatorisch istAWS Management Console:

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster und dann die Instance aus.
3. Wählen Sie Wartung aus.
4. Suchen Sie im Abschnitt Ausstehende Wartung das Betriebssystem-Update und überprüfen Sie den Statuswert.

In der ist der Wartungsstatus AWS Management Consoleeines Betriebssystem-Updates auf „Verfügbar“ gesetzt und es gibt kein Anwendungsdatum, wie in der folgenden Abbildung dargestellt:



Sie können das Betriebssystemupdate auswählen und im Abschnitt Ausstehende Wartung auf Jetzt anwenden oder Beim nächsten Wartungsfenster anwenden klicken. Wenn der Wartungswert das nächste Fenster ist, verschieben Sie die Wartungselemente, indem Sie Upgrade verschieben auswählen. Sie können eine Wartungsaktion nicht verschieben, wenn sie bereits gestartet wurde.

Alternativ können Sie die Instance aus einer Liste von Clustern auswählen, indem Sie im Navigationsbereich auf Cluster klicken, und im Menü Aktionen die Option Jetzt anwenden oder Beim nächsten Wartungsfenster anwenden auswählen.

Using the AWS CLI

Rufen Sie den Befehl auf, um zu ermittelnAWS CLI, ob ein Update optional oder obligatorisch ist describe-pending-maintenance-actions :

```
aws docdb describe-pending-maintenance-actions
```

Ein obligatorisches Betriebssystem-Update enthält einen `AutoAppliedAfterDate`- und einen `CurrentApplyDate`-Wert. Ein optionales Betriebssystem-Update enthält diese Werte nicht.

Die folgende Ausgabe zeigt ein obligatorisches Betriebssystem-Update:

```
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
The following output shows an optional operating system update.
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Verfügbarkeit von Betriebssystem-Updates

Betriebssystem-Updates sind spezifisch für Amazon DocumentDB-Engine-Versionen und Instance-Klassen. Daher erhalten oder erfordern Amazon DocumentDB-Instances Aktualisierungen zu verschiedenen Zeiten. Wenn für Ihre Instance basierend auf der Engine-Version und Instance-Klasse ein Betriebssystem-Update verfügbar ist, wird das Update in der Konsole angezeigt. Sie kann auch durch Ausführen des AWS CLI `describe-pending-maintenance-actions` Befehls oder durch Aufrufen der `DescribePendingMaintenanceActions` API-Operation angezeigt werden. Wenn ein Update für Ihre Instance verfügbar ist, können Sie Ihr Betriebssystem aktualisieren, indem Sie den Anweisungen unter Anwenden von Amazon DocumentDB-Updates folgen.

Grundlegendes zu serviceverknüpften Rollen

Amazon DocumentDB (mit MongoDB-Kompatibilität) verwendet serviceverknüpfte AWS Identity and Access Management Rollen (IAM). Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon DocumentDB verknüpft ist. Serviceverknüpfte Rollen sind von Amazon

DocumentDB vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle erleichtert die Verwendung von Amazon DocumentDB, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon DocumentDB definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon DocumentDB seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dadurch werden Ihre Amazon DocumentDB-Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Services, die servicegebundene Rollen unterstützen, finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Servicegebundene Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Berechtigungen für serviceverknüpfte Amazon DocumentDB-Rollen

Amazon DocumentDB (mit MongoDB-Kompatibilität) verwendet die serviceverknüpfte Rolle, die so benannt ist AWSServiceRoleForRDS, dass Amazon DocumentDB AWS Dienste im Namen Ihrer Cluster aufrufen kann.

Die serviceverknüpfte Rolle AWSServiceRoleForRDS vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `docdb.amazonaws.com`

Die Richtlinie für Rollenberechtigungen ermöglicht Amazon DocumentDB, die folgenden Aktionen für die angegebenen Ressourcen auszuführen:

- Aktionen auf `ec2`:
  - `AssignPrivateIpAddresses`
  - `AuthorizeSecurityGroupIngress`
  - `CreateNetworkInterface`
  - `CreateSecurityGroup`

  - `DeleteNetworkInterface`
  - `DeleteSecurityGroup`
  - `DescribeAvailabilityZones`
  - `DescribeInternetGateways`
  - `DescribeSecurityGroups`
  - `DescribeSubnets`
  - `DescribeVpcAttribute`
  - `DescribeVpcs`
  - `ModifyNetworkInterfaceAttribute`
  - `RevokeSecurityGroupIngress`
  - `UnassignPrivateIpAddresses`
- Aktionen auf `sns`:
  - `ListTopic`
  - `Publish`
- Aktionen auf `cloudwatch`:
  - `PutMetricData`
  - `GetMetricData`
  - `CreateLogStream`
  - `PullLogEvents`
  - `DescribeLogStreams`
  - `CreateLogGroup`

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Möglicherweise erhalten Sie die folgende Fehlermeldung:
> Unable to create the resource. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später noch einmal.
> Wenn Sie diesen Fehler erhalten, stellen Sie sicher, dass Sie die folgenden Berechtigungen aktiviert haben:

```
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"rds.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Eine serviceverknüpfte Amazon DocumentDB-Rolle erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen Cluster erstellen, erstellt Amazon DocumentDB die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und dann erneut erstellen müssen, können Sie die Rolle in Ihrem Konto mit demselben Verfahren neu anlegen. Wenn Sie einen Cluster erstellen, erstellt Amazon DocumentDB die serviceverknüpfte Rolle erneut für Sie.

Ändern einer serviceverknüpften Amazon DocumentDB-Rolle

Amazon DocumentDB erlaubt es Ihnen nicht, die AWSServiceRoleForRDS serviceverknüpfte Rolle zu ändern. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können die Beschreibung der Rolle jedoch mithilfe von IAM ändern. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Amazon DocumentDB-Rolle

Wenn Sie ein Feature oder einen Service, die bzw. der eine servicegebundene Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch alle Ihre -Cluster löschen, bevor Sie die serviceverknüpfte Rolle löschen können.

Bereinigen einer mit dem Amazon DocumentDB-Dienst verknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie über die Konsole, ob die servicegebundene Rolle eine aktive Sitzung hat:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich der IAM-Konsole Roles und dann den Namen (nicht das Kontrollkästchen) der AWSServiceRoleForRDSRolle aus.
3. Wählen Sie auf der Seite Summary für die ausgewählte Rolle die Registerkarte Access Advisor.
4. Überprüfen Sie auf der Registerkarte Access Advisor (Advisor aufrufen) die jüngsten Aktivitäten für die serviceverknüpfte Rolle.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob Amazon DocumentDB die AWSServiceRoleForRDS Rolle verwendet, können Sie versuchen, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt die Löschung fehl und Sie können die -Regionen anzeigen, in denen die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Wenn Sie die Rolle AWSServiceRoleForRDS entfernen wollen, müssen Sie zunächst alle Instances und Cluster löschen. Informationen zum Löschen von Instances und Clustern finden Sie in den folgenden Themen:

- Löschen einer Amazon DocumentDB-Instance
- Löschen eines Amazon DocumentDB-Clusters

Unterstützte Regionen für serviceverknüpfte Amazon DocumentDB-Rollen

Amazon DocumentDB unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter https://

docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html#regions-and-azs-availability.

Verwenden von elastischen Amazon DocumentDB-Clustern

Elastische Amazon DocumentDB-Cluster unterstützen Workloads mit Millionen von Lese-/Schreibvorgängen pro Sekunde und Petabyte Speicherkapazität. Elastic Cluster vereinfachen auch die Interaktion von Entwicklern mit Amazon DocumentDB, da keine Instances ausgewählt, verwaltet oder aktualisiert werden müssen.

Elastische Amazon DocumentDB-Cluster wurden erstellt, um:

- Stellen Sie eine Lösung für Kunden bereit, die nach einer Datenbank suchen, die praktisch unbegrenzte Skalierung mit umfangreichen Abfragefunktionen und MongoDB-API-Kompatibilität bietet.
- Geben Sie Kunden höhere Verbindungslimits und reduzieren Sie Ausfallzeiten durch Patches.
- Investieren Sie weiterhin in eine cloudnative, elastische und klassenführerische Architektur für JSON-Workloads.

Themen

Anwendungsfälle für Elastic Cluster

Dokumentdatenbanken sind nützlich für Workloads, die ein flexibles Schema für eine schnelle, iterative Entwicklung benötigen. Beispiele für Amazon DocumentDB-Anwendungsfälle finden Sie unter Anwendungsfälle der Dokumentdatenbank.

Im Folgenden finden Sie einige Beispiele für Anwendungsfälle, für die elastische Cluster erhebliche Vorteile bieten können:

Benutzerprofile

Da Dokumentdatenbanken über ein flexibles Schema verfügen, können sie Dokumente mit unterschiedlichen Attributen und Datenwerten in großem Umfang speichern. Elastic Cluster sind eine praktische Lösung für Online-Profile, in denen verschiedene Benutzer verschiedene Arten von Informationen bereitstellen. Angenommen, Ihre Anwendungen unterstützen Hunderte von Millionen von Benutzerprofilen. Sie können elastische Cluster verwenden, um solche Anwendungen zu unterstützen, da sie hoch- und hochskaliert werden können, um Millionen von Schreib- und Lesevorgängen für diese Benutzerprofile zu unterstützen. Sie können auch außerhalb der Spitzenzeiten herunterskalieren, um die Kosten zu senken.

Inhaltsverwaltung und historische Datensätze

Um Inhalte effektiv zu verwalten, müssen Sie in der Lage sein, Inhalte aus einer Vielzahl von Quellen zu sammeln, zu aggregieren und dann an den Client zu liefern. Aufgrund ihres flexiblen Schemas sind Dokumentdatenbanken ideal für die Erfassung und Speicherung jeglicher Art von Daten.
Sie können sie verwenden, um neue Inhaltstypen zu erstellen und zu integrieren, einschließlich benutzergenerierter Inhalte wie Bilder, Kommentare und Videos. Im Laufe der Zeit benötigt Ihre Datenbank möglicherweise mehr Speicherplatz. Mit elastischen Clustern können Sie Ihre Daten auf mehr Speicher-Volumes verteilen, sodass Sie Petabyte an Daten in einem einzigen Cluster speichern können.

Vorteile von Elastic Clustern

AWS -Serviceintegration

Elastische Amazon DocumentDB-Cluster lassen sich genauso in andere - AWS Services integrieren wie Amazon DocumentDB:

- Migration – Sie können AWS Database Migration Service (DMS) verwenden, um von MongoDB und anderen relationalen Datenbanken zu elastischen Amazon DocumentDB-Clustern zu migrieren.
- Überwachung – Sie können den Zustand und die Leistung Ihres elastischen Clusters mit Amazon überwachen CloudWatch.
- Sicherheit – Sie können Authentifizierung und Autorisierung über AWS Identity and Access Management (IAM) einrichten, um Ihre elastischen Cluster zu verwalten und Amazon VPC für sichere reine VPC-Verbindungen zu verwenden.
- Datenverwaltung – Sie können verwenden AWS Glue , um Daten von/zu anderen - AWS Services wie Amazon S3, Amazon Redshift und Amazon OpenSearch Service zu importieren und zu exportieren.

Verfügbarkeit von Elastic Cluster-Regionen und -Versionen

Verfügbarkeit in Regionen

Die folgende Tabelle zeigt die AWS Regionen, in denen elastische Amazon DocumentDB-Cluster derzeit verfügbar sind, und den Endpunkt für jede Region.

Name der Region	Region	Availability Zones
USA Ost (Nord-Virginia)	`us-east-1`	5
USA Ost (Ohio)	`us-east-2`	3
USA West (Oregon)	`us-west-2`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	3
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Südamerika (São Paulo)	`sa-east-1`	3

Name der Region	Region	Availability Zones
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3

Verfügbarkeit von Versionen

Elastic Cluster unterstützen das mit MongoDB 5.0 kompatible Wire-Protokoll. Unterschiede zwischen Instance-basierten DocumentDB-4.0-Clustern und elastischen Clustern finden Sie unter Funktionsunterschiede zwischen Amazon DocumentDB 4.0 und elastischen Clustern.

Einschränkungen

Elastic-Cluster-Verwaltung

Die folgenden Cluster-Verwaltungsfunktionen und -funktionen werden in dieser Version nicht unterstützt:

- Möglichkeit, globale Cluster zu erstellen
- Vorhandene Amazon DocumentDB-Ereignisse und Abonnieren von Ereignissen
- Bereichs-Sharding
- Vorhandene Shard-Sammlung
- Shard-Schlüssel mit mehreren Feldern
- Shard-Schlüssel ändern
- Point-in-time -Wiederherstellung
- Klonen
- Performance Insights

> **Note**
>
> Weitere Informationen zu Elastic-Cluster-Limits finden Sie unter Kontingente und Limits für Amazon DocumentDB .

Abfrage- und Schreibvorgänge

Die folgenden Abfrage- und Schreiboperationsbefehle und -funktionen werden in dieser Version nicht unterstützt:

- DDL-Befehle während Skalierungsvorgängen
- Profiler
- Parametergruppen
- AWS Config
- AWS Backup

Sammlungs- und Indexverwaltung

Die folgenden Sammlungs- und Indexverwaltungsfunktionen werden in dieser Version nicht unterstützt:

- Sparse Indexes
- Geodatenindizes
- Hintergrundindex erstellen

Administration und Diagnose

Die folgenden Administrations- und Diagnosebefehle und -funktionen werden in dieser Version nicht unterstützt:

- AWS Secrets Manager
- Benutzerdefinierte R ole-based-access-control (RBAC)-Rollen.
- Beim Herstellen einer Verbindung wird das Schreibproblem 0 nicht unterstützt.
- Ändern von Subnetzen, die zu einer VPC gehören, die derzeit keinem vorhandenen elastischen Cluster zugewiesen ist.

Anmeldefunktionen

Die folgenden Amazon DocumentDB-Opt-In-Funktionen werden in dieser Version nicht unterstützt:

- ACID-Transaktionen
- DDL/DML-Prüfung
- Change streams
- Sitzungsbefehle

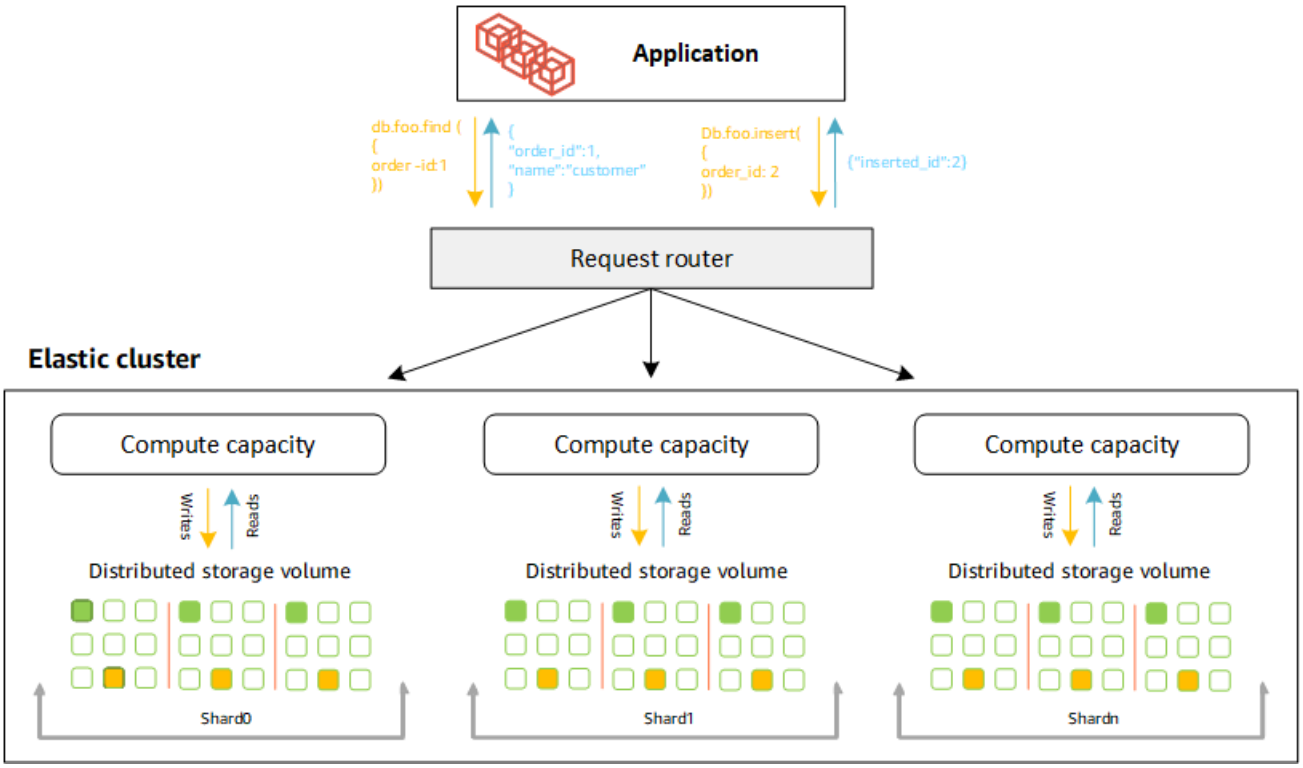
Elastische Amazon DocumentDB-Cluster: Funktionsweise

Die Themen in diesem Abschnitt enthalten Informationen zu den Mechanismen und Funktionen, die Amazon DocumentDB-Elastic-Cluster unterstützen.

Themen

Amazon DocumentDB-Cluster-Sharding

Elastische Amazon DocumentDB-Cluster verwenden Hash-basiertes Sharding, um Daten über verteilte Speichersysteme zu partitionieren. Sharding, auch bekannt als Partitionierung, teilt große Datensätze in kleine Datensätze auf mehrere Knoten auf, sodass Sie Ihre Datenbank über vertikale Skalierungsgrenzen hinaus skalieren können. Elastic Cluster verwenden die Trennung oder „Entkopplung“ von Rechenleistung und Speicher in Amazon DocumentDB, sodass Sie unabhängig voneinander skalieren können. Anstatt Sammlungen durch Verschieben kleiner Datenblöcke zwischen Datenverarbeitungsknoten neu zu partitionieren, kopieren elastische Cluster Daten effizient innerhalb des verteilten Speichersystems.



Shard-Definitionen

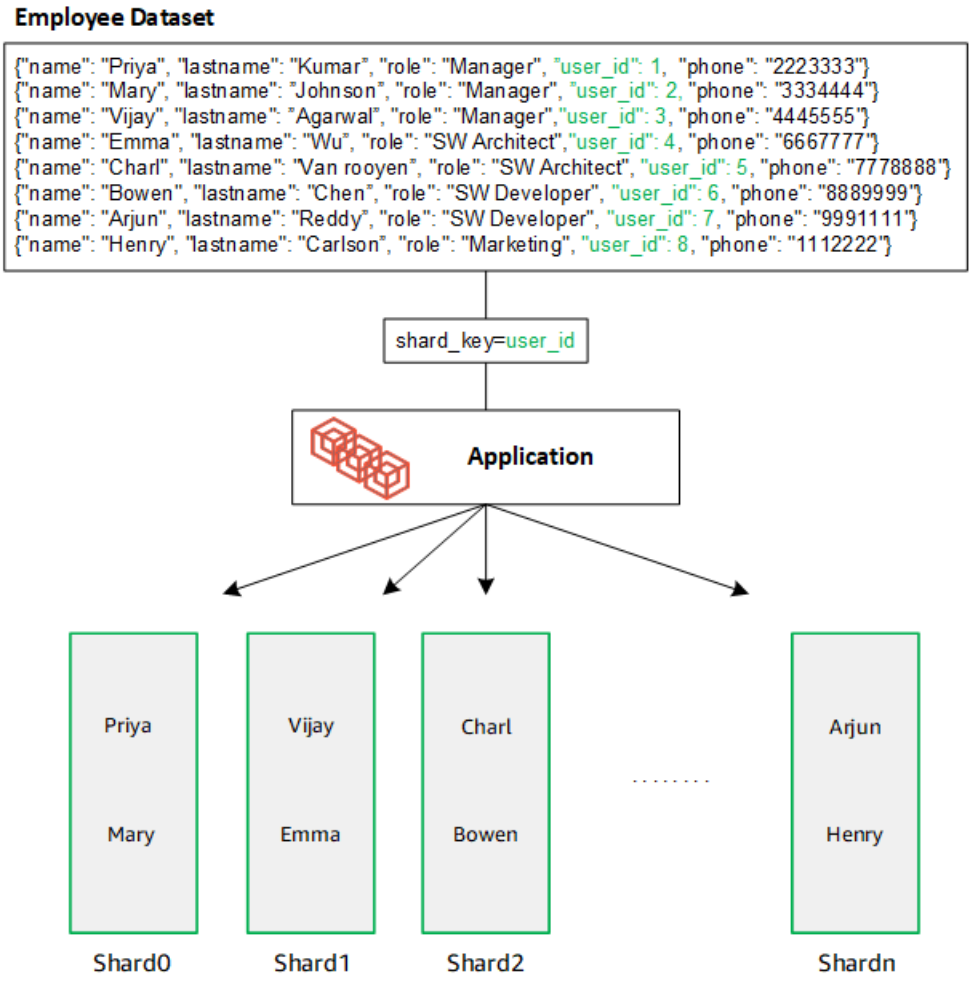
Definitionen der Shard-Knotenklatur:

- Shard – Ein Shard bietet Rechenleistung für einen elastischen Cluster. Ein Shard hat standardmäßig zwei Knoten. Sie können maximal 32 Shards konfigurieren und jeder Shard kann maximal 64 vCPUs haben.
- Shard-Schlüssel – Ein Shard-Schlüssel ist ein Pflichtfeld in Ihren JSON-Dokumenten in Sharded-Sammlungen, die Elastic Cluster verwenden, um Lese- und Schreibdatenverkehr an den übereinstimmenden Shard zu verteilen.
- Shard-Sammlung – Eine Shard-Sammlung ist eine Sammlung, deren Daten in Datenpartitionen über einen elastischen Cluster verteilt werden.
- Partition – Eine Partition ist ein logischer Teil der Sharded-Daten. Wenn Sie eine Sharded-Sammlung erstellen, werden die Daten basierend auf dem Shard-Schlüssel automatisch in Partitionen innerhalb jedes Shards organisiert. Jeder Shard hat mehrere Partitionen.

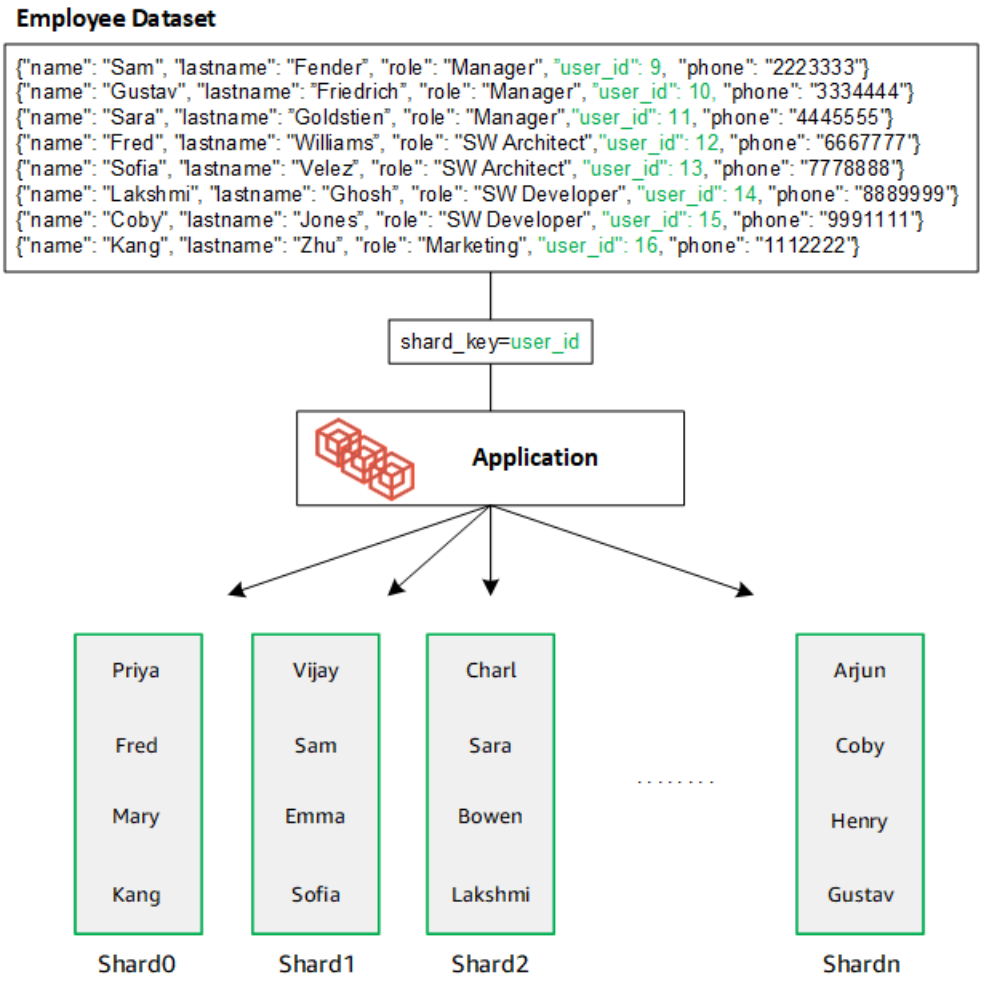
Verteilen von Daten auf konfigurierte Shards

Erstellen Sie einen Shard-Schlüssel mit vielen eindeutigen Werten. Ein guter Shard-Schlüssel partitioniert Ihre Daten gleichmäßig auf die zugrunde liegenden Shards und gibt Ihrem Workload den

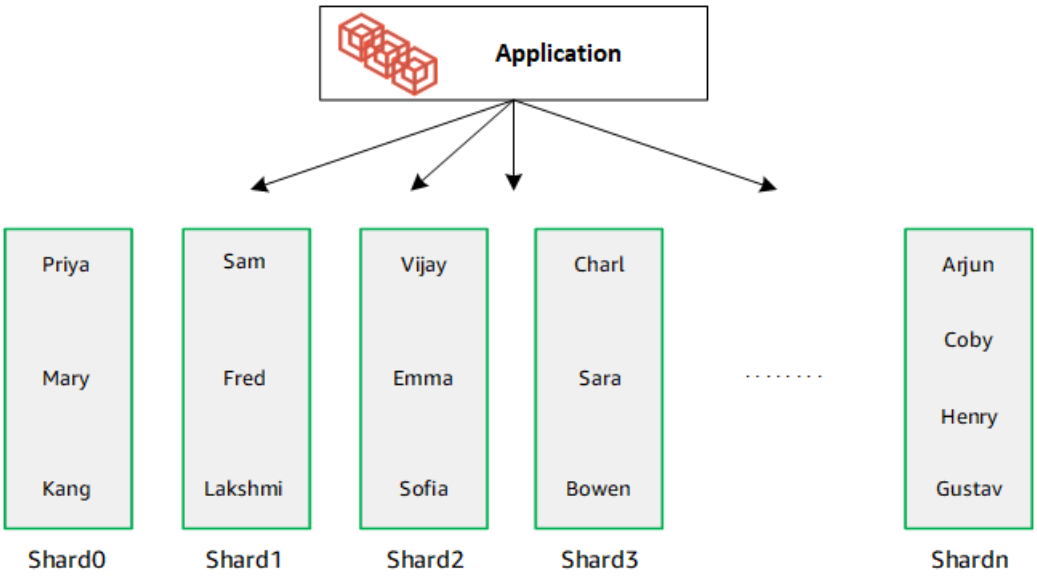
besten Durchsatz und die beste Leistung. Das folgende Beispiel zeigt Mitarbeiternamendaten, die einen Shard-Schlüssel mit dem Namen „user_id“ verwenden:



DocumentDB verwendet Hash-Sharding, um Ihre Daten auf zugrunde liegende Shards zu partitionieren. Additonale Daten werden auf die gleiche Weise eingefügt und verteilt:



Wenn Sie Ihre Datenbank durch Hinzufügen zusätzlicher Shards aufskalieren, verteilt Amazon DocumentDB die Daten automatisch neu:



Elastic Cluster-Migration

Amazon DocumentDB unterstützt die Migration von MongoDB-Sharded-Daten zu elastischen Clustern. Offline-, Online- und Hybrid-Migrationsmethoden werden unterstützt. Weitere Informationen finden Sie unter Migration zu Amazon DocumentDB.

Elastic Cluster-Skalierung

Amazon DocumentDB Elastic Clusters bietet die Möglichkeit, die Anzahl der Shards (Aufskalierung) in Ihrem elastischen Cluster und die Anzahl der vCPUs zu erhöhen, die auf jeden Shard angewendet werden (Aufskalierung). Sie können auch die Anzahl der Shards und die Rechenkapazität (vCPUs) nach Bedarf reduzieren.

Bewährte Methoden für die Skalierung finden Sie unter SkalPartischlüsselN.

> **Note**
>
> Die Skalierung auf Clusterebene ist ebenfalls verfügbar. Weitere Informationen finden Sie unter Skalieren von Amazon DocumentDB-Clustern.

Zuverlässigkeit des Elastic Clusters

Amazon DocumentDB ist auf Zuverlässigkeit, Beständigkeit und Fehlertoleranz ausgelegt. Um die Verfügbarkeit zu verbessern, stellen elastische Cluster zwei Knoten pro Shard bereit, die sich über verschiedene Availability Zones befinden. Amazon DocumentDB enthält mehrere automatische Funktionen, die es zu einer zuverlässigen Datenbanklösung machen. Weitere Informationen finden Sie unter Zuverlässigkeit von Amazon DocumentDB .

Speicher und Verfügbarkeit von Elastic Clustern

Amazon DocumentDB-Daten werden in einem Cluster-Volume gespeichert. Dabei handelt es sich um ein einzelnes virtuelles Volume, das Solid-State-Laufwerke (SSDs) verwendet. Ein Cluster-Volume besteht aus sechs Kopien Ihrer Daten, die automatisch über mehrere Availability Zones in einer einzigen AWS Region repliziert werden. Diese Replikation trägt dazu bei, dass Ihre Daten sehr langlebig sind und weniger Datenverlust möglich ist. Sie trägt außerdem dazu bei, dass Ihr Cluster während eines Failovers besser verfügbar ist, da Kopien Ihrer Daten bereits in anderen Availability Zones vorhanden sind. Weitere Informationen zu Speicher, Hochverfügbarkeit und Replikation finden Sie unter Amazon DocumentDB: Funktionsweise.

Funktionsunterschiede zwischen Amazon DocumentDB 4.0 und elastischen Clustern

Die folgenden funktionalen Unterschiede bestehen zwischen Amazon DocumentDB 4.0 und elastischen Clustern.

- Die Ergebnisse von `top` und `collStats` werden nach Shards partitioniert.
- Sammlungsstatistiken von `top` und `collStats` für Sharded-Sammlungen werden zurückgesetzt, wenn die Anzahl der Cluster-Shards geändert wird.
- Die integrierte Backup-Rolle unterstützt jetzt `serverStatus`. Aktion – Entwickler und Anwendungen mit Sicherungsrolle können Statistiken über den Status des Amazon DocumentDB-Clusters sammeln.
- Das `SecondaryDelaySecs` Feld wird `slaveDelay` in der `replSetGetConfig` Ausgabe ersetzt.
- Der `hello` Befehl ersetzt `isMaster` – `hello` gibt ein Dokument zurück, das die Rolle des elastischen Clusters beschreibt.
- Der `$elemMatch` Operator in elastischen Clustern gleicht nur Dokumente auf der ersten Verschachtelungsebene eines Arrays ab. In Amazon DocumentDB 4.0 durchläuft der Operator alle Ebenen, bevor übereinstimmende Dokumente zurückgegeben werden. Beispielsweise:

```
db.foo.insert(
[
    {a: {b: 5}},
    {a: {b: [5]}},
    {a: {b: [3, 7]}},
    {a: [{b: 5}]},
    {a: [{b: 3}, {b: 7}]},
    {a: [{b: [5]}]},
    {a: [{b: [3, 7]}]},
    {a: [[{b: 5}]]},
    {a: [[{b: 3}, {b: 7}]]},
    {a: [[{b: [5]}]]},
    {a: [[{b: [3, 7]}]]}
]);
// Elastic Clusters
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
```

```
// Docdb 4.0: traverse more than one level deep
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }
```

- Die Projektion „$“ in Amazon DocumentDB 4.0 gibt alle Dokumente mit allen Feldern zurück. Bei elastischen Clustern gibt der `find` Befehl mit der Projektion „$“ Dokumente zurück, die mit dem Abfrageparameter übereinstimmen, der nur das Feld enthält, das mit der Projektion „$“ übereinstimmt.
- In elastischen Clustern geben die `find` Befehle mit den `$options` Abfrageparametern `$regex` und den folgenden Fehler zurück: „Optionen können nicht sowohl in $regex als auch in $options festgelegt werden“.

- Bei elastischen Clustern gibt `$indexOfCP` jetzt „-1“ zurück, wenn:
  - die Teilzeichenfolge nicht in der gefunden wird`string expression`, oder
  - `start` ist eine Zahl größer als oder `end`
  - `start` ist eine Zahl, die größer als die Bytelänge der Zeichenfolge ist.

  In Amazon DocumentDB 4.0 gibt „0“ `$indexOfCP` zurück, wenn die `start` Position eine Zahl größer als `end` oder die Bytelänge der Zeichenfolge ist.
- Bei elastischen Clustern geben Projektionsoperationen in `_id fields`beispielsweise: Dokumente zurück`{"_id.nestedField" : 1}`, die nur das projizierte Feld enthalten. Während in Amazon DocumentDB 4.0 verschachtelte Feldprojektionsbefehle kein Dokument herausfiltern.

Erste Schritte mit elastischen Amazon DocumentDB-Clustern

In diesem Abschnitt „Erste Schritte“ erfahren Sie, wie Sie Ihren ersten elastischen Cluster erstellen und abfragen können. Es gibt viele Möglichkeiten, eine Verbindung herzustellen und mit Elastic Clustern zu beginnen. Dieses Handbuch verwendet AWS Cloud9, ein webbasiertes Terminal, um Ihren elastischen Cluster mithilfe der mongo-Shell direkt aus dem zu verbinden und abzufragen AWS Management Console.

Themen

- Einrichten
- Schritt 1: Erstellen eines elastischen Clusters

- Schritt 2: Erstellen einer AWS Cloud9 Umgebung
- Schritt 3: Installieren der mongo-Shell
- Schritt 4: Herstellen einer Verbindung mit Ihrem neuen elastischen Cluster
- Schritt 5: Sharden Ihrer Sammlung; Einfügen und Abfragen von Daten

Einrichten

Wenn Sie lieber von Ihrem lokalen Computer aus eine Verbindung zu Ihrem Amazon DocumentDB herstellen möchten, indem Sie eine SSH-Verbindung zu einer Amazon EC2-Instance herstellen, finden Sie weitere Informationen unter Herstellen einer Verbindung mit Amazon EC2.

Voraussetzungen

Bevor Sie Ihren ersten Amazon DocumentDB-Cluster erstellen, müssen Sie Folgendes tun:

Erstellen eines Amazon Web Services (AWS)-Kontos

Bevor Sie Amazon DocumentDB verwenden können, benötigen Sie ein Amazon Web Services (AWS)-Konto. Das AWS Konto ist kostenlos. Sie zahlen nur für die Services und Ressourcen, die Sie wirklich nutzen.

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um eines zu erstellen.

So registrieren Sie sich für ein AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für ein registrieren AWS-Konto, Root-Benutzer des AWS-Kontos wird ein erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Richten Sie die erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen ein.

Für den Zugriff auf die Verwaltung von Amazon DocumentDB-Ressourcen wie Cluster, Instances und Cluster-Parametergruppen sind Anmeldeinformationen erforderlich, die zur Authentifizierung Ihrer Anforderungen verwenden AWS kann. Weitere Informationen finden Sie unter Identity and Access Management für Amazon DocumentDB.

1. Geben Sie in der Suchleiste von IAM AWS Management Consoleein und wählen Sie IAM im Dropdown-Menü aus.
2. Sobald Sie sich in der IAM-Konsole befinden, wählen Sie im Navigationsbereich Benutzer aus.
3. Wählen Sie Ihren Benutzernamen aus.
4. Klicken Sie auf die Schaltfläche Berechtigungen hinzufügen .
5. Wählen Sie die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.
6. Geben Sie `AmazonDocDBFullAccess` in die Suchleiste ein und wählen Sie sie aus, sobald sie in den Suchergebnissen erscheint.
7. Klicken Sie unten auf die blaue Schaltfläche, auf der Weiter: Überprüfen steht.
8. Klicken Sie unten auf die blaue Schaltfläche mit der Meldung Berechtigungen hinzufügen .

Erstellen einer Amazon Virtual Private Cloud (Amazon VPC)

Dieser Schritt ist nur erforderlich, wenn Sie noch keine Standard-Amazon-VPC haben. Wenn Sie dies nicht tun, führen Sie Schritt 1 von Erste Schritte mit Amazon VPC im Amazon-VPC-Benutzerhandbuch aus. Dies dauert weniger als fünf Minuten.

Schritt 1: Erstellen eines elastischen Clusters

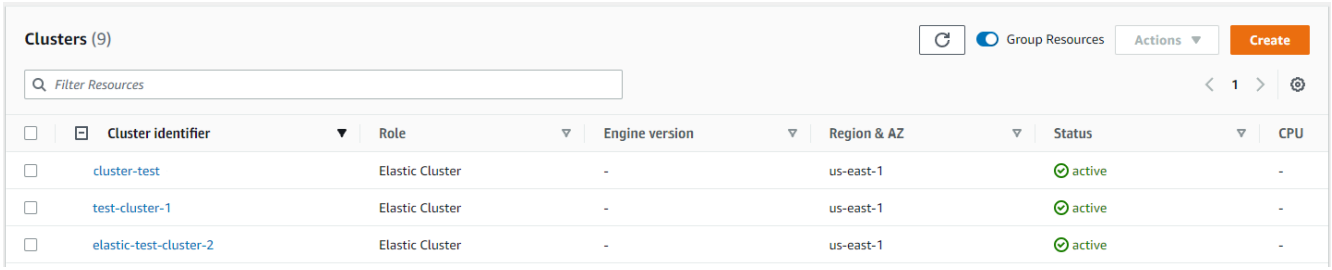
In diesem Abschnitt wird erläutert, wie Sie einen völlig neuen elastischen Cluster mit der AWS Management Console oder AWS CLI mit den folgenden Anweisungen erstellen.

Using the AWS Management Console

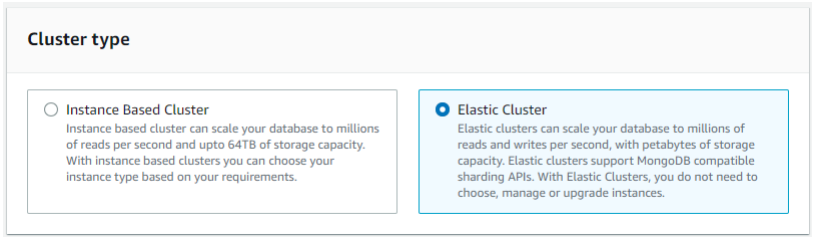
So erstellen Sie eine Elastic-Cluster-Konfiguration mit der AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.

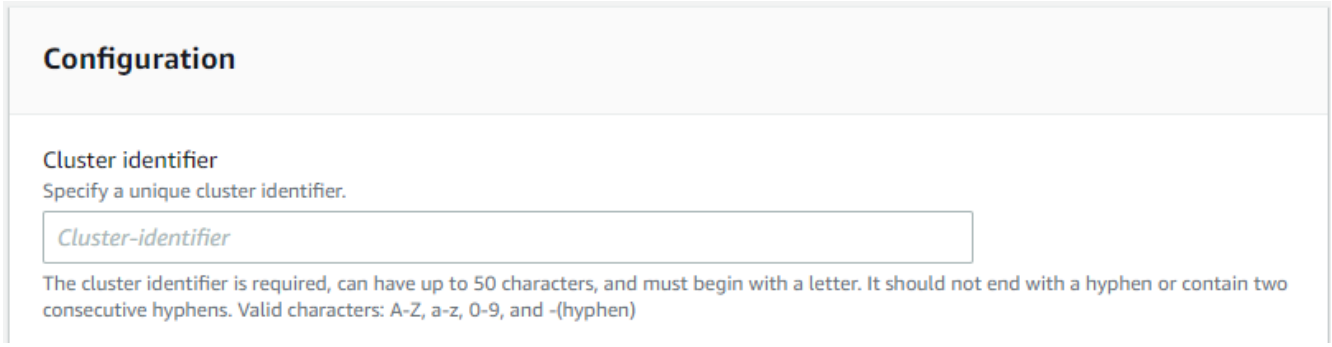
2. Wählen Sie in der Amazon DocumentDB-Managementkonsole unter Cluster die Option Erstellen aus.



3. Wählen Sie auf der Seite Amazon DocumentDB-Cluster erstellen im Abschnitt Clustertyp die Option Elastic Cluster aus.



4. Geben Sie auf der Seite Amazon DocumentDB-Cluster erstellen im Abschnitt Konfiguration eine eindeutige Cluster-ID ein (gemäß den Benennungsanforderungen unter dem Feld).



5. Für die Shard-Konfigurationsfelder:

   a. Geben Sie im Feld Shard-Anzahl die Anzahl der Shards ein, die Sie in Ihrem Cluster haben möchten. Die maximale Anzahl von Shards pro Cluster beträgt 32.

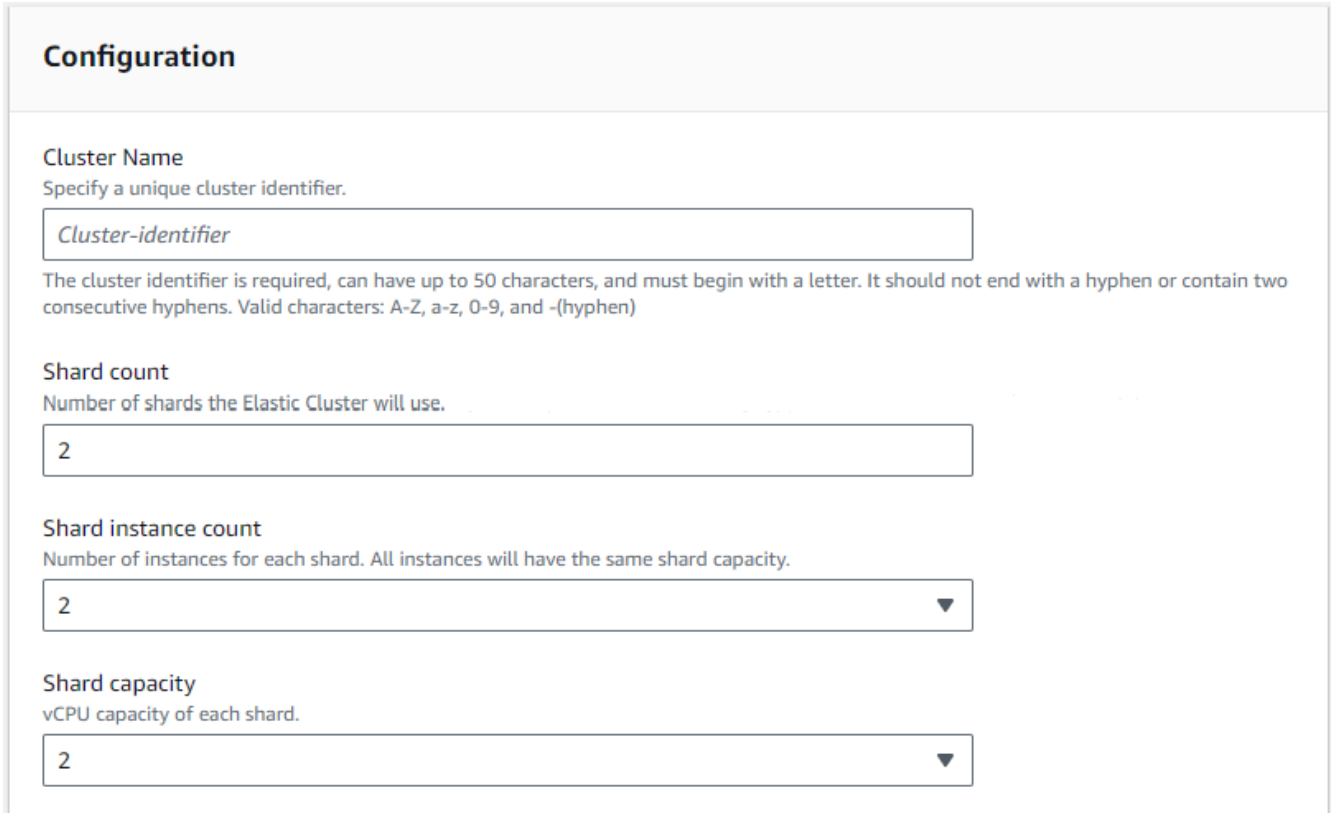
   > **Note**
   >
   > Für jeden Shard werden zwei Knoten bereitgestellt. Beide Knoten haben dieselbe Shard-Kapazität.

   b. Wählen Sie im Feld Anzahl der Shard-Instances die Anzahl der Replikat-Instances aus, die jedem Shard zugeordnet werden sollen. Die maximale Anzahl von Shard-Instances

beträgt 16 in Schritten von 1. Alle Replikat-Instances haben die gleiche Shard-Kapazität, wie im folgenden Feld definiert.

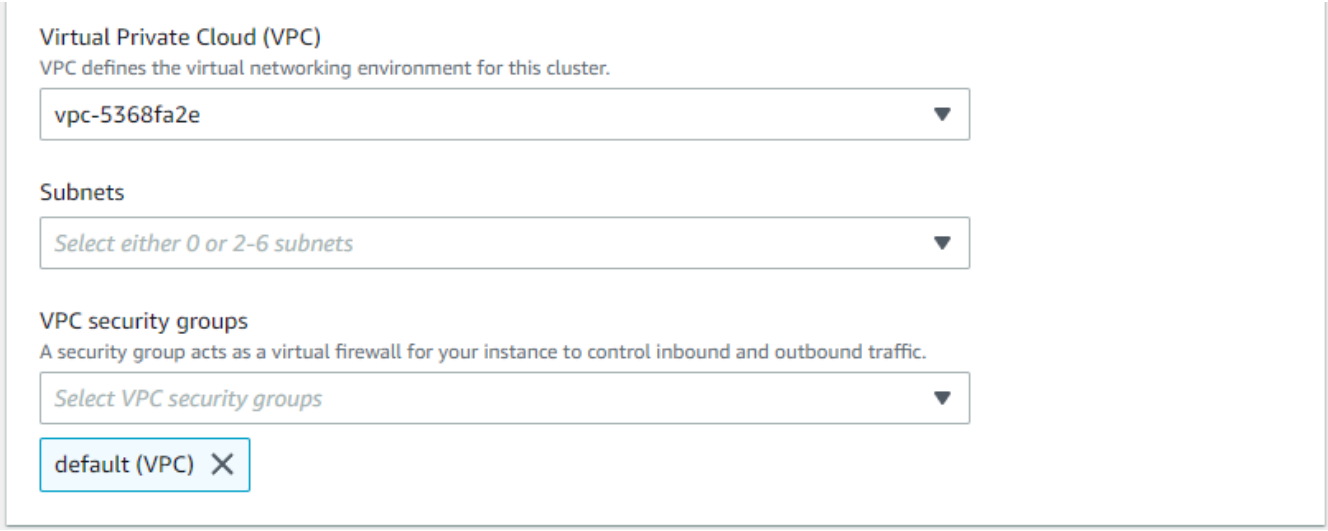
> **Note**
>
> Die Anzahl der Replikat-Instances gilt für alle Shards im elastischen Cluster. Ein Wert für die Anzahl der Shard-Instances von 1 bedeutet, dass es eine Writer-Instance gibt und alle zusätzlichen Instances Replikate sind, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

c. Wählen Sie im Feld Shard-Kapazität die Anzahl der virtuellen CPUs (vCPUs), die Sie jeder Shard-Instance zuordnen möchten. Die maximale Anzahl von vCPUs pro Shard-Instance beträgt 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.



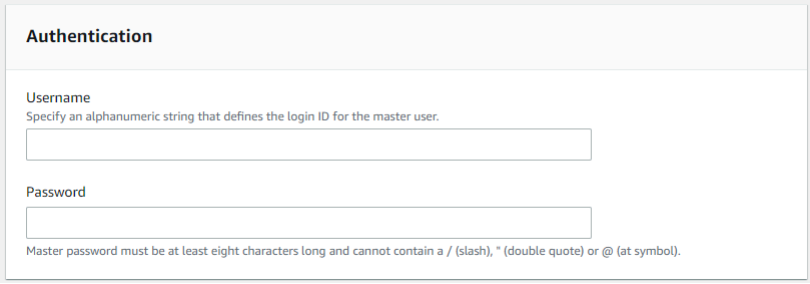
6. Wählen Sie im Feld Virtual Private Cloud (VPC) eine VPC aus der Dropdown-Liste aus.

Für Subnetze und VPC-Sicherheitsgruppen können Sie die Standardeinstellungen verwenden oder drei Subnetze Ihrer Wahl und bis zu drei VPC-Sicherheitsgruppen (mindestens eine) auswählen.



7. Geben Sie im Abschnitt Authentifizierung eine Zeichenfolge ein, die den Anmeldenamen des Hauptbenutzers im Feld Benutzername identifiziert.

   Geben Sie im Feld Passwort ein eindeutiges Passwort ein, das den Anweisungen entspricht.

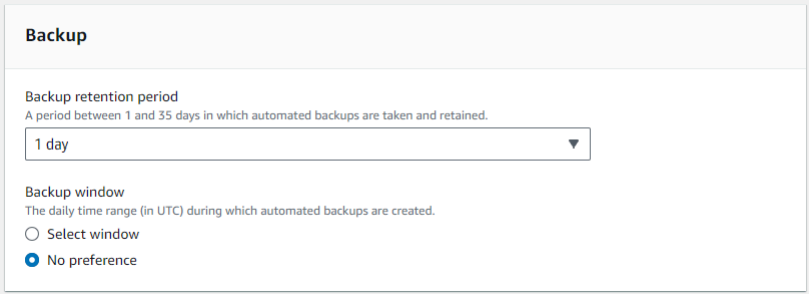


8. Behalten Sie im Abschnitt Verschlüsselung die Standardeinstellungen bei.

   Optional können Sie einen von Ihnen erstellten AWS KMS key ARN eingeben. Weitere Informationen finden Sie unter Datenverschlüsselung im Ruhezustand im Ruhezustand und Deaktivieren Datenverschlüsselung Amazon DocumentDB DB-Cluster.

   > **Important**
   >
   > Die Verschlüsselung muss für Elastic Cluster aktiviert sein.

9. Bearbeiten Sie im Abschnitt Backup die Felder entsprechend Ihren Backup-Anforderungen.



a. Aufbewahrungszeitraum für Backups – Wählen Sie in der Liste die Anzahl der Tage aus, für die automatische Backups dieses Clusters aufbewahrt werden sollen, bevor sie gelöscht werden.

b. Backup-Fenster – Legen Sie die tägliche Zeit und Dauer fest, in der Amazon DocumentDB Backups dieses Clusters erstellen soll.

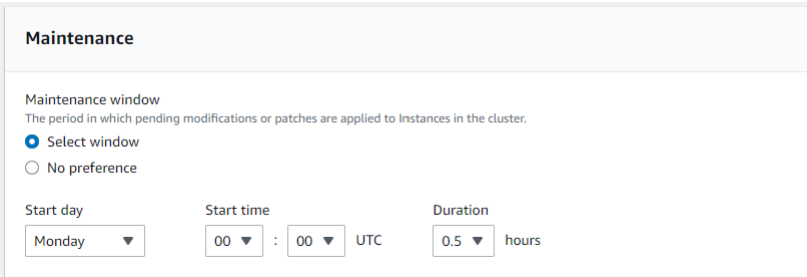
   i. Wählen Sie Fenster auswählen, wenn Sie die Zeit und Dauer der Erstellung von Sicherungen konfigurieren möchten.

      Startzeit – Wählen Sie in der ersten Liste die Startzeitstunde (UTC) für den Start Ihrer automatischen Backups aus. Wählen Sie in der zweiten Liste die Minute für den Beginn der automatischen Backups aus.

      Dauer – Wählen Sie in der Liste die Anzahl der Stunden aus, die dem Erstellen automatischer Backups zugewiesen werden sollen.

   ii. Wählen Sie Keine Präferenz, wenn Amazon DocumentDB die Zeit und Dauer der Erstellung von Backups auswählen soll.

10. Wählen Sie im Abschnitt Wartung den Tag, die Uhrzeit und die Dauer aus, an denen Änderungen oder Patches auf Ihren Cluster angewendet werden.



11. Wählen Sie Cluster erstellen.

Der elastische Cluster wird jetzt bereitgestellt. Dies kann bis zu einigen Minuten dauern. Sie können eine Verbindung zu Ihrem Cluster herstellen, wenn der Elastic-Cluster-Status als **`active`** in der Cluster-Liste angezeigt wird.

Using the AWS CLI

Um einen elastischen Cluster mit der zu erstellen AWS CLI, verwenden Sie die `-create-cluster`Operation mit den folgenden Parametern:

- --cluster-name—Erforderlich. Der aktuelle Name des Clusters für elastische Skalierung, wie er während der Erstellung oder letzten Änderung eingegeben wurde.
- --shard-capacity—Erforderlich. Die Anzahl der vCPUs, die jedem Shard zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.
- --shard-count—Erforderlich. Die Anzahl der dem Cluster zugewiesenen Shards. Das Maximum ist 32.
- --shard-instance-count—Optional. Die Anzahl der Replikat-Instances, die für alle Shards in diesem Cluster gelten. Das Maximum ist 16.
- --admin-user-name—Erforderlich. Der Benutzername, der dem Administratorbenutzer zugeordnet ist.
- --admin-user-password—Erforderlich. Das Passwort, das dem Administratorbenutzer zugeordnet ist.
- --auth-type—Erforderlich. Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das Passwort abgerufen werden soll, das für den Zugriff auf den elastischen Cluster verwendet wird. Gültige Typen sind `PLAIN_TEXT` oder `SECRET_ARN`.
- --vpc-security-group-ids—Optional. Konfigurieren Sie eine Liste von EC2-VPC-Sicherheitsgruppen, die diesem Cluster zugeordnet werden sollen.
- --preferred-maintenance-window—Optional. Konfigurieren Sie den wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

  Das Format ist: `ddd:hh24:mi-ddd:hh24:mi`. Gültige Tage (TT): Mo, Di, Mi, Do, Fr, Sa, So

  Die Standardeinstellung ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock für jede Amazon-Web-Services-Region an einem zufälligen Wochentag ausgewählt wird.

  Mindestfenster von 30 Minuten.

- --kms-key-id—Optional. Konfigurieren Sie die KMS-Schlüsselkennung für einen verschlüsselten Cluster.

  Die KMS-Schlüsselkennung ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon Web Services-Konto erstellen, das den KMS-Verschlüsselungsschlüssel besitzt, der zum Verschlüsseln des neuen Clusters verwendet wird, können Sie den KMS-Schlüssel-Alias anstelle des ARN für den KMS-Verschlüsselungsschlüssel verwenden.

  Wenn in kein Verschlüsselungsschlüssel angegeben ist KmsKeyId und der `StorageEncrypted` Parameter „true“ ist, verwendet Amazon DocumentDB Ihren Standardverschlüsselungsschlüssel.
- --preferred-backup-window—Optional. Der täglich bevorzugte Zeitraum, in dem automatische Backups erstellt werden. Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock für jede ausgewählt wird AWS-Region.
- --backup-retention-period—Optional. Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Der Standardwert lautet 1.
- --storage-encrypted—Optional. Konfiguriert, ob der Cluster verschlüsselt ist oder nicht.

  `--no-storage-encrypted` gibt an, dass der Cluster nicht verschlüsselt ist.
- --subnet-ids—Optional. Konfigurieren Sie Netzwerksubnetz-IDs.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

> **Note**
>
> Die folgenden Beispiele umfassen die Erstellung eines bestimmten KMS-Schlüssels. Um den Standard-KMS-Schlüssel zu verwenden, schließen Sie den `--kms-key-id` Parameter nicht ein.

Für Linux, macOS oder Unix:

```
aws docdb-elastic create-cluster \
     --cluster-name sample-cluster-123 \
     --shard-capacity 8 \
     --shard-count 4 \
```

```
    --shard-instance-count 3 \
    --auth-type PLAIN_TEXT \
    --admin-user-name testadmin \
    --admin-user-password testPassword \
    --vpc-security-group-ids ec-65f40350 \
    --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \
    --preferred-backup-window 18:00-18:30 \
    --backup-retention-period 7
```

Für Windows:

```
aws docdb-elastic create-cluster ^
    --cluster-name sample-cluster-123 ^
    --shard-capacity 8 ^
    --shard-count 4 ^
    --shard-instance-count 3 ^
    --auth-type PLAIN_TEXT ^
    --admin-user-name testadmin ^
    --admin-user-password testPassword ^
    --vpc-security-group-ids ec-65f40350 ^
    --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \
    --preferred-backup-window 18:00-18:30 \
    --backup-retention-period 7
```

Schritt 2: Erstellen einer AWS Cloud9 Umgebung

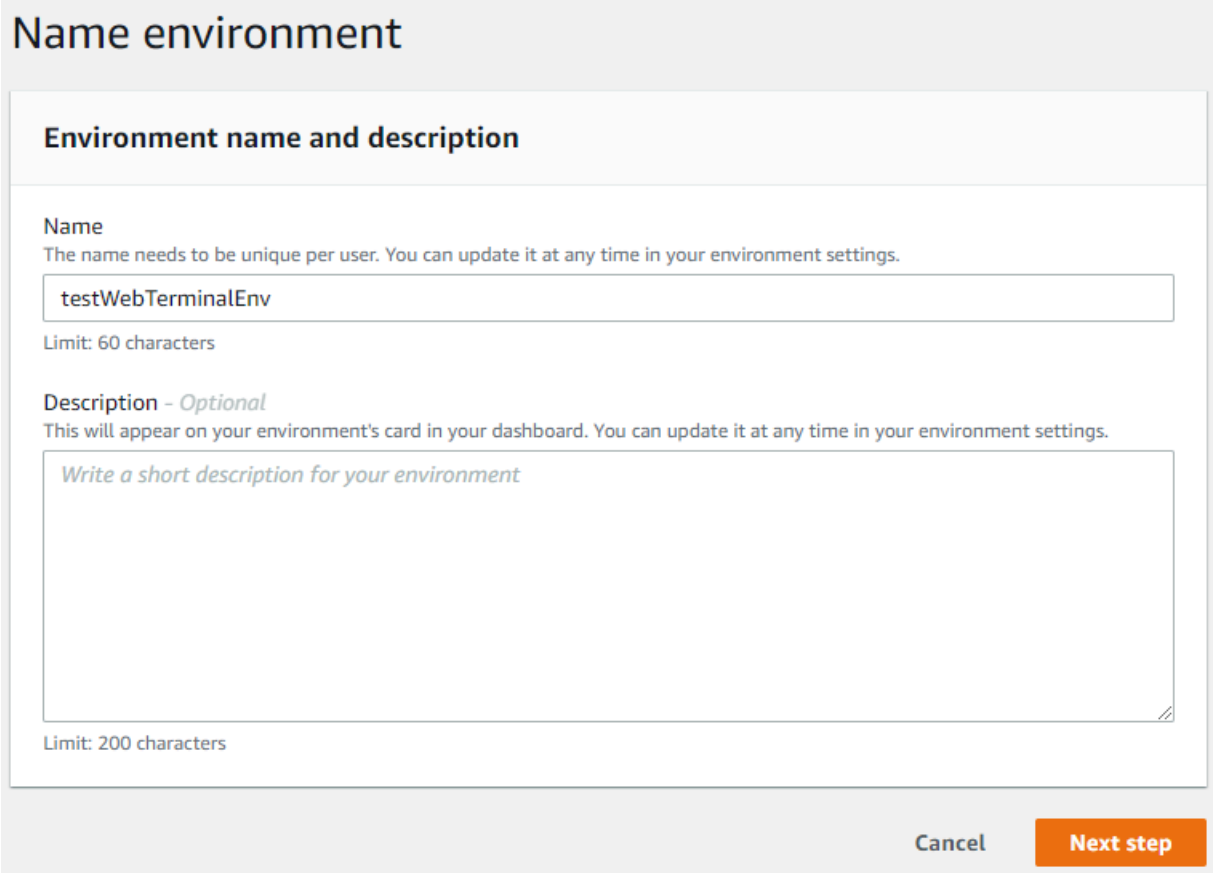
AWS Cloud9 bietet ein webbasiertes Terminal, mit dem Sie mithilfe der mongo-Shell eine Verbindung zu Ihren elastischen Amazon DocumentDB-Clustern herstellen und diese abfragen können.

> **Note**
>
> Hinweis: Ihre AWS Cloud9 Umgebung muss sich in derselben Sicherheitsgruppe wie Ihre Instance befinden. Sie können die Sicherheitsgruppe in der Amazon EC2-Konsole ändern.

1. Verwenden Sie Ihr - AWS Konto und greifen Sie auf zuAWS Management Console.

2. Navigieren Sie zur AWS Cloud9 Konsole . Sie können „Cloud9“ in das Suchfeld eingeben, um es zu finden.
3. Wählen Sie auf der AWS Cloud9 Umgebungsstartseite Umgebung erstellen aus.
4. Geben Sie auf der Seite Nameumgebung im Feld Name einen Namen Ihrer Wahl ein.

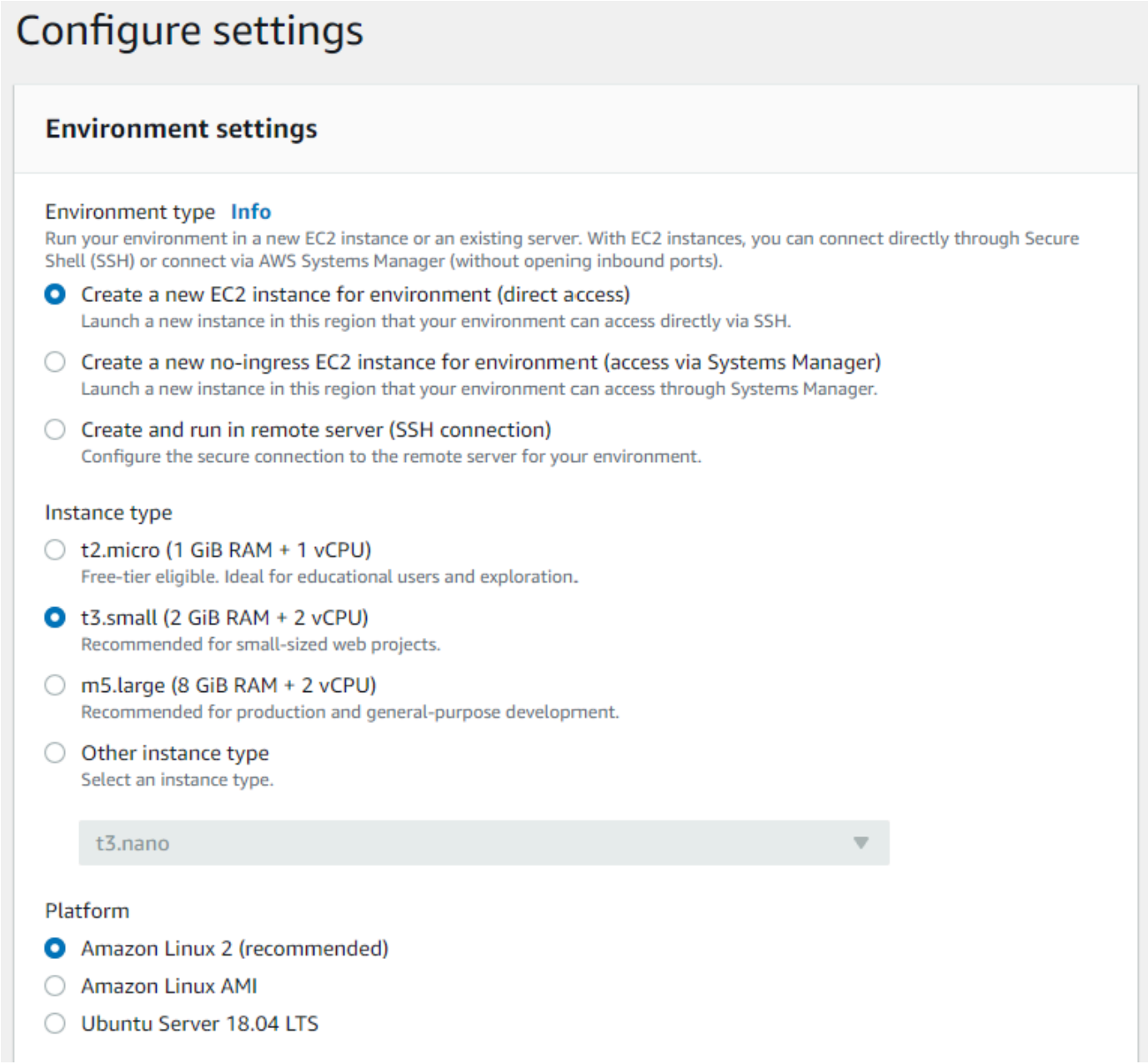
   Klicken Sie auf Nächster Schritt.



5. Wählen Sie unter Umgebungseinstellungen im Abschnitt Umgebungstyp die Option Neue EC2-Instance für die Umgebung erstellen (direkter Zugriff) aus.

   Wählen Sie im Abschnitt Instance-Typ einen geeigneten Instance-Typ für Ihr Netzwerk aus.

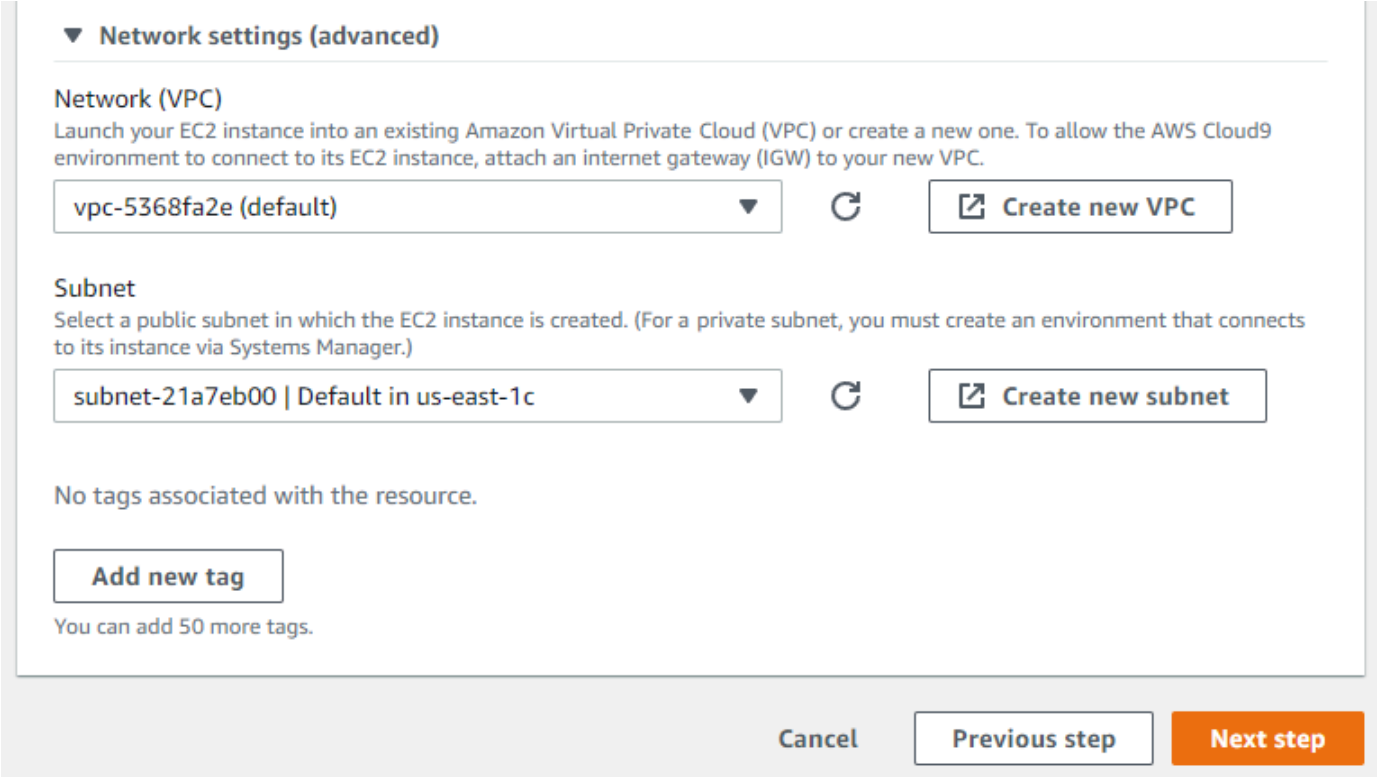
   Wählen Sie im Abschnitt Plattform die Option Amazon Linux 2 aus (empfohlen).



6. Erweitern Sie Network settings (advanced) (Netzwerkeinstellungen (erweitert)).

   Wählen Sie die VPC und eines der Subnetze aus, die Sie beim Erstellen Ihres elastischen Clusters verwendet haben.

   Klicken Sie auf Nächster Schritt.



7. Überprüfen Sie Ihre AWS Cloud9 Konfiguration.

   Wenn Ihre Konfiguration korrekt ist, wählen Sie Umgebung erstellen aus.

Schritt 3: Installieren der mongo-Shell

Sobald Ihre AWS Cloud9 Umgebung bereit ist, können Sie eine Verbindung zu Ihrem Cluster herstellen. Als Nächstes installieren Sie die mongo-Shell in Ihrer AWS Cloud9 Umgebung, die Sie in Schritt 3 erstellt haben. Die mongo-Shell ist ein Befehlszeilendienstprogramm, mit dem Sie Ihren elastischen Cluster verbinden und abfragen können.

Wenn Ihre AWS Cloud9 Umgebung ab Schritt 3 noch offen ist, kehren Sie zu dieser Umgebung zurück und fahren Sie mit Anweisung 3 fort. Wenn Sie die AWS Cloud9 Umgebung verlassen haben, suchen Sie in der AWS Cloud9 Konsole unter Ihre Umgebungen die Umgebung mit dem Namen, den Sie im vorherigen Schritt festgelegt haben. Wählen Sie IDE öffnen aus.

1. Erstellen Sie an der Eingabeaufforderung die Repository-Datei mit dem folgenden Befehl:

Example

```
echo -e "[mongodb-org-4.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2013.03/mongodb-org/4.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-4.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-4.0.repo
```

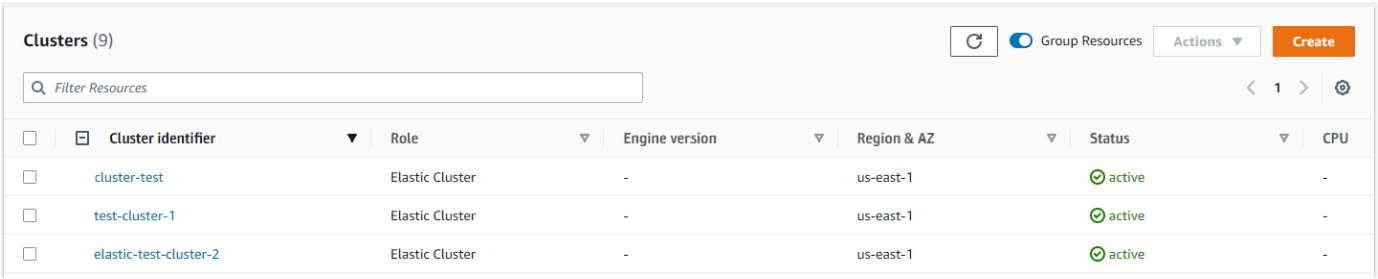
2. Wenn der Vorgang abgeschlossen ist, installieren Sie die mongo-Shell mit dem folgenden Befehl:

   ```
   sudo yum install -y mongodb-org-shell
   ```

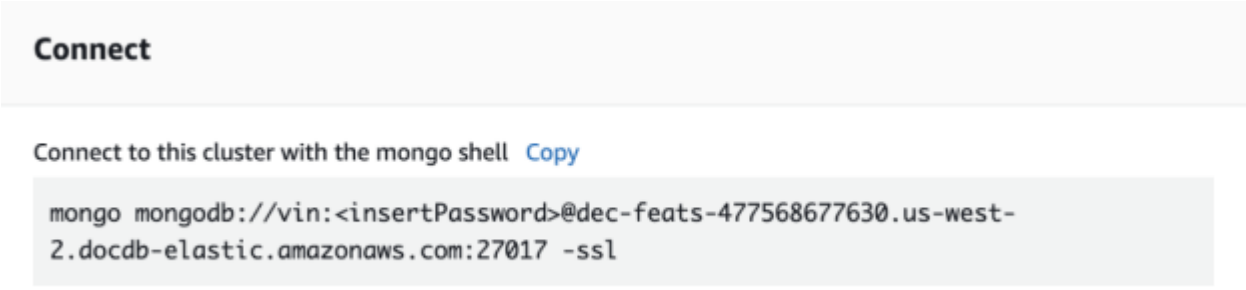
Schritt 4: Herstellen einer Verbindung mit Ihrem neuen elastischen Cluster

Stellen Sie mithilfe der mongo-Shell, die Sie in Schritt 4 installiert haben, eine Verbindung zu Ihrem Cluster her.

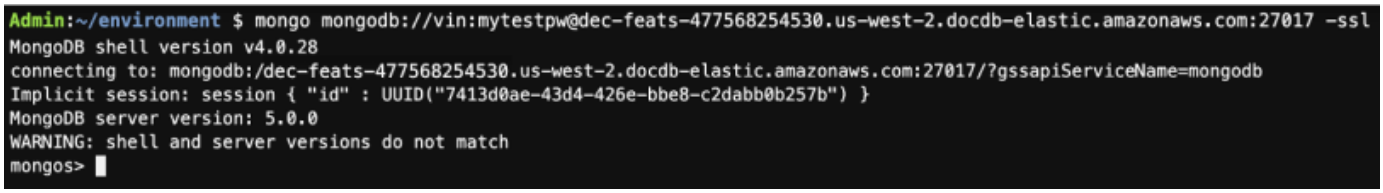
1. Suchen Sie in der Amazon DocumentDB-Managementkonsole unter Cluster Ihren Cluster. Sortieren Sie nach Rolle, um alle Cluster mit der Rolle Elastic Cluster anzuzeigen.



2. Wählen Sie den Cluster aus, den Sie erstellt haben, indem Sie die Cluster-ID auswählen. Kopieren Sie unter Konnektivität und Sicherheit Ihren Endpunkt und fügen Sie ihn in Ihre AWS Cloud9 Umgebung ein.



3. Nach der Verbindung sollte die folgende Ausgabe angezeigt werden:



Schritt 5: Sharden Ihrer Sammlung; Einfügen und Abfragen von Daten

Elastic Cluster bieten Unterstützung für Sharding in Amazon DocumentDB . Nachdem Sie nun mit Ihrem Cluster verbunden sind, können Sie den Cluster fragmentieren, Daten einfügen und einige Abfragen ausführen.

1. Um eine Sammlung zu fragmentieren, geben Sie Folgendes ein:

   ```
   sh.shardCollection("db.Employee1" , { "Employeeid" : "hashed" })
   ```

2. Um ein einzelnes Dokument einzufügen, geben Sie Folgendes ein:

   ```
   db.Employee1.insert({"Employeeid":1, "Name":"Joe", "LastName": "Bruin",
   "level": 1 })
   ```

   Die folgende Ausgabe wird angezeigt:

   ```
   WriteResult({ "nInserted" : 1 })
   ```

3. Um das von Ihnen geschriebene Dokument zu lesen, geben Sie den `findOne()` Befehl ein (es wird ein einzelnes Dokument zurückgegeben):

   ```
   db.Employee1.findOne()
   ```

   Die folgende Ausgabe wird angezeigt:

   Example

   ```
   {
   "_id" : ObjectId("61f344e0594fe1a1685a8151"),
   "EmployeeID" : 1,
   "Name" : "Joe",
   "LastName" : "Bruin",
   "level" : 1
   }
   ```

4. Um einige weitere Abfragen durchzuführen, sollten Sie einen Anwendungsfall für das Gaming-Profil in Betracht ziehen. Fügen Sie zunächst einige Einträge in eine Sammlung mit dem Namen „Mitarbeiter“ ein. Geben Sie Folgendes ein:

   Example

   ```
   db.Employee1.insertMany([
       { "Employeeid" : 1, "name" : "Matt", "lastname": "Winkle", "level": 12},
       { "Employeeid" : 2, "name" : "Frank", "lastname": "Chen", "level": 2},
       { "Employeeid" : 3, "name" : "Karen", "lastname": "William", "level": 7},
       { "Employeeid" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}
   ])
   ```

   Die folgende Ausgabe wird angezeigt:

   ```
   { "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
   ```

5. Um alle Dokumente in der Profilsammlung zurückzugeben, geben Sie den Befehl `find`() ein:

   ```
   db.Employee1.find()
   ```

   Die Daten, die Sie in Schritt 4 eingegeben haben, werden angezeigt.

6. Um ein einzelnes Dokument abzufragen, fügen Sie einen Filter ein (z. B.: „Katie“). Geben Sie Folgendes ein:

   ```
   db.Employee1.find({name: "Katie"})
   ```

   Die folgende Ausgabe wird angezeigt:

   ```
   { "_id" : 4, "name" : "Katie", "lastname": "Schaper", "level": 3}
   ```

7. Um ein Profil zu finden und es zu ändern, geben Sie den `findAndModify` Befehl ein. In diesem Beispiel erhält der Mitarbeiter „Matt“ ein höheres Level von „14“:

   Example

   ```
   db.Employee1.findAndModify({
       query: { "Employeeid" : 1, "name" : "Matt"},
       update: { "Employeeid" : 1, "name" : "Matt", "lastname" : "Winkle", "level" :
    14 }
   })
   ```

Die folgende Ausgabe wird angezeigt (beachten Sie, dass sich die Ebene noch nicht geändert hat):

Example

```
{
    "_id" : 1,
    "name" : "Matt",
    "lastname" : "Winkle",
    "level" : 12,
}
```

8. Um die Erhöhung der Ebene zu überprüfen, geben Sie die folgende Abfrage ein:

   ```
   db.Employee1.find({name: "Matt"})
   ```

   Die folgende Ausgabe wird angezeigt:

   ```
   { "_id" : 1, "name" : "Matt", "lastname" : "winkle", "level" : 14 }
   ```

Bewährte Methoden

Informieren Sie sich über die bewährten Methoden für die Arbeit mit Amazon DocumentDB Docschlüsseln. Alle bewährten Methoden für instanzbasierte Amazon DocumentDB-Cluster gelten auch für elastische Cluster. Dieser Abschnitt wird fortlaufend aktualisiert, wenn neue bewährte Methoden identifiziert werden.

Themen

- AusPartiN
- Verbindungsverwaltung
- Ungeteilte Sammlungen
- SkalPartischlüsselN
- ÜberPartischlüsselN

AusPartiN

In der folgenden Liste werden Richtlinien für die Erstellung von Shard-Schlüsseln beschrieben.

- Verwenden Sie einen gleichmäßig verteilten Hash-Schlüssel, um Ihre Daten auf alle Shards in Ihrem Cluster zu verteilen (vermeiden Sie Tastenkombinationen).
- Verwenden Sie Ihren Shard-Schlüssel in allen Lese-/Aktualisierungs- und Löschanfragen, um Scatter-Gather-Abfragen zu vermeiden.
- Vermeiden Sie verschachtelte Shard-Schlüssel, wenn Sie Lese-, Aktualisierungs- und Löschvorgänge ausführen.
- Wenn Sie Batch-Operationen ausführen, setzen Sie`ordered` den Wert auf false, damit alle Shards parallel ausgeführt werden können und die Latenzen verbessert werden.

Verbindungsverwaltung

In der folgenden Liste werden Richtlinien für die Verwaltung Ihrer Verbindungen zu Ihrer Datenbank beschrieben.

- Überwachen Sie die Anzahl Ihrer Verbindungen und wie oft neue Verbindungen geöffnet und geschlossen werden.
- Verteilen Sie Ihre Verbindungen auf alle Subnetze in der Konfiguration Ihrer Anwendung. Wenn Ihr Cluster in mehreren Subnetzen konfiguriert ist, Sie aber nur eine Teilmenge der Subnetze nutzen, kann es zu Engpässen bei der maximalen Anzahl von Verbindungen kommen.

Ungeteilte Sammlungen

Im Folgenden wird eine Richtlinie für Sammlungen ohne Sharded beschrieben.

- Wenn Sie mit unsharded Collections arbeiten, versuchen Sie zur Lastverteilung, häufig genutzte unsharded Collections in verschiedenen Datenbanken zu behalten. Elastische Amazon DocumentDB-Cluster platzieren Datenbanken auf verschiedenen Shards und speichern unshardierte Sammlungen für dieselbe Datenbank auf demselben Shard.

SkalPartischlüsselN

In der folgenden Liste werden Richtlinien für die Skalierung Ihrer elastischen Cluster beschrieben.

- Skalierungsvorgänge können kurzzeitig zu zeitweiligen Datenbank- und Netzwerkfehlern führen. Vermeiden Sie nach Möglichkeit eine Skalierung während der Spitzenzeiten. Informieren Sie sich über die Arbeit mit der SkalschlüsselN.

- Das Hoch- und Herunterskalieren der Shard-Kapazität (Änderung der vCPU-Anzahl pro Shard) zur Erhöhung der Rechenleistung wird einer Erhöhung oder Verringerung der Shard-Anzahl vorgezogen, da dies schneller ist und intermittierende Datenbank- und Netzwerkfehler eine kürzere Dauer haben.
- Wenn Sie ein Wachstum erwarten, ziehen Sie es vor, die Anzahl der Shards zu erhöhen, anstatt die Shard-Kapazität zu skalieren. Auf diese Weise können Sie Ihren Cluster skalieren, indem Sie die Shard-Kapazität für Szenarien erhöhen, in denen Sie schnell skalieren müssen.
- Überwachen Sie Ihre clientseitigen Wiederholungsrichtlinien und versuchen Sie es erneut mit exponentiellem Backoff und Jitter, um eine Überlastung Ihrer Datenbank zu vermeiden, wenn bei der Skalierung Fehler auftreten.

ÜberPartischlüsselN

In der folgenden Liste werden Richtlinien für die Überwachung Ihrer elastischen Cluster beschrieben.

- Verfolge daspeak-to-average Verhältnis deiner Metriken pro Shard, um festzustellen, ob dein Traffic ungleichmäßig ist (verwende einen Hotkey/Hotspot). Die wichtigsten Kennzahlen zur Erfassung derpeak-to-average Kennzahlen sind:
  - `PrimaryInstanceCPUUtilization`
    - Dies kann auf der Ebene pro Shard überwacht werden.
    - Auf Clusterebene können Sie den durchschnittlichen Skew bis p99 überwachen.
  - `PrimaryInstanceFreeableMemory`
    - Dies kann auf der Ebene pro Shard überwacht werden.
    - Auf Clusterebene können Sie den durchschnittlichen Skew bis p99 überwachen.
  - `DatabaseCursorsMax`
    - Dies sollte auf der Ebene pro Shard überwacht werden, um den Skew zu ermitteln.
  - `Documents-Inserted/Updated/Returned/Deleted`
    - Dies sollte auf der Ebene pro Shard überwacht werden, um den Skew zu ermitteln.

Verwalten elastischer Cluster

Um einen elastischen Amazon DocumentDB-Cluster zu verwalten, benötigen Sie eine IAM-Richtlinie mit den entsprechenden Berechtigungen auf Steuerebene von Amazon DocumentDB. Mit diesen Berechtigungen können Sie Cluster erstellen, ändern und löschen. Die Amazon

DocumentDBFullAccess -Richtlinie stellt alle erforderlichen Berechtigungen für die Verwaltung eines elastischen Amazon DocumentDB-Clusters bereit.

Die folgenden Themen zeigen, wie Sie verschiedene Aufgaben ausführen, wenn Sie mit elastischen Amazon DocumentDB-Clustern arbeiten.

Themen

Ändern von Elastic-Cluster-Konfigurationen

In diesem Abschnitt wird erläutert, wie Sie elastische Cluster mithilfe der AWS Management Console oder AWS CLI mit den folgenden Anweisungen ändern.

Eine primäre Verwendung der Änderung des Clusters besteht darin, Shards zu skalieren, indem die Shard-Anzahl und/oder die Shard-Rechenkapazität erhöht oder verringert werden.

Using the AWS Management Console

So ändern Sie eine Elastic-Cluster-Konfiguration mithilfe der AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Cluster-ID den Namen des Clusters aus, den Sie ändern möchten.
4. Wählen Sie Ändern aus.

5. Bearbeiten Sie die Felder, die Sie ändern möchten, und wählen Sie dann Cluster ändern aus.

Configuration

Cluster identifier

SampleCluster

Shard count

Number of shards the Elastic Cluster will use.

2

Shard instance count

Number of instances for each shard. All instances will have the same shard capacity.

2

Shard capacity

vCPU capacity of each shard.

2

Maintenance

Maintenance window

The period in which pending modifications or patches are applied to your Elastic cluster.

- ○ Select window
- ◉ No preference

Authentication

Username

SampleUser

New password

Confirm new password

Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).

Network settings

Subnets

Select either 0 or 2-6 subnets

subnet-0b2962f92a0f5a8fb ✕ subnet-08c6d849efd4dfe96 ✕

VPC security groups

> **Note**
>
> Alternativ können Sie auf das Dialogfeld Cluster ändern zugreifen, indem Sie die Seite Cluster aufrufen, das Kontrollkästchen neben Ihrem Cluster aktivieren, Aktionen auswählen und dann Ändern auswählen.

Using the AWS CLI

Um eine Elastic-Cluster-Konfiguration mit der zu ändern AWS CLI, verwenden Sie die `-update-cluster`Operation mit den folgenden Parametern:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, den Sie ändern möchten.
- **`--shard-capacity`**—Optional. Die Anzahl der vCPUs, die jedem Shard zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.
- **`--shard-count`**—Optional. Die Anzahl der dem Cluster zugewiesenen Shards. Das Maximum ist 32.
- **`--shard-instance`**-count – Optional. Die Anzahl der Replikat-Instances, die für alle Shards in diesem Cluster gelten. Das Maximum ist 16.
- **`--auth-type`**—Optional. Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das Passwort abgerufen werden soll, das für den Zugriff auf den elastischen Cluster verwendet wird. Gültige Typen sind `PLAIN_TEXT` oder `SECRET_ARN`.
- **`--admin-user-password`**—Optional. Das Passwort, das dem Administratorbenutzer zugeordnet ist.
- **`--vpc-security-group-ids`**—Optional. Konfigurieren Sie eine Liste von Amazon EC2- und Amazon Virtual Private Cloud (VPC)-Sicherheitsgruppen, die diesem Cluster zugeordnet werden sollen.
- **`--preferred-maintenance-window`**—Optional. Konfigurieren des wöchentlichen Zeitraums, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time)

  Das Format ist: `ddd:hh24:mi-ddd:hh24:mi`. Gültige Tage (TT): Mo, Di, Mi, Do, Fr, Sa, So

  Die Standardeinstellung ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock für jede Amazon-Web-Services-Region an einem zufälligen Wochentag ausgewählt wird.

Mindestfenster von 30 Minuten.

- **`--subnet-ids`**—Optional. Konfigurieren Sie Netzwerksubnetz-IDs.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic update-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
    --shard-capacity 8 \
    --shard-count 4 \
    --shard-instance-count 3 \
    --admin-user-password testPassword \
    --vpc-security-group-ids ec-65f40350 \
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Für Windows:

```
aws docdb-elastic update-cluster ^
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
    --shard-capacity 8 ^
    --shard-count 4 ^
    --shard-instance-count 3 ^
    --admin-user-password testPassword ^
    --vpc-security-group-ids ec-65f40350 ^
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Informationen zum Überwachen des Status des elastischen Clusters nach Ihrer Änderung finden Sie unter Überwachen eines elastischen Clusters.

Überwachung eines elastischen Clusters

In diesem Abschnitt wird erläutert, wie Sie Ihren elastischen Cluster mithilfe der AWS Management Console oder AWS CLI mit den folgenden Anweisungen überwachen.

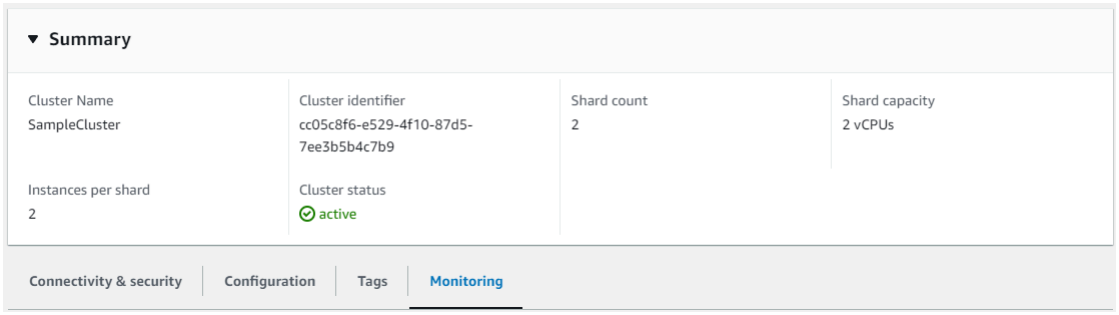
Using the AWS Management Console

So überwachen Sie eine Elastic-Cluster-Konfiguration mit der AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Cluster-ID den Namen des Clusters aus, den Sie überwachen möchten.
4. Wählen Sie die Registerkarte Überwachung.



Für die folgenden Überwachungskategorien CloudWatch wird eine Reihe von Diagrammen von Amazon angezeigt:

- Ressourcenauslastung
- Durchsatz
- Latency
- Operationen
- System (System)

Sie können auch CloudWatch über die auf Amazon zugreifen AWS Management Console , um Ihre eigene Überwachungsumgebung für Ihre elastischen Cluster einzurichten.

Using the AWS CLI

Um eine bestimmte Elastic-Cluster-Konfiguration mit der zu überwachen AWS CLI, verwenden Sie die -`get-cluster`Operation mit den folgenden Parametern:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, für den Sie Informationen benötigen.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic get-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Für Windows:

```
aws docdb-elastic get-cluster ^
    --cluster-arn arn:aws:docdb:-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Die Ausgabe dieser Operation sieht etwa wie folgt aus:

```
"cluster": {
        ...
        "clusterArn": "arn:aws:docdb-elastic:us-
west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9",
        "clusterEndpoint": "stretch-11-477568257630.us-east-1.docdb-
elastic.amazonaws.com",
        "readerEndpoint": "stretch-11-477568257630-ro.us-east-1.docdb-
elastic.amazonaws.com",
        "clusterName": "stretch-11",
        "shardCapacity": 2,
        "shardCount": 3,
        "shardInstanceCount: 5,
        "status": "ACTIVE",
        ...
 }
```

Weitere Informationen finden Sie unter `DescribeClusterSnapshot` in der API-Referenz zur Amazon DocumentDB-Ressourcenverwaltung.

Um die Details aller elastischen Cluster mit der anzuzeigen AWS CLI, verwenden Sie die `-list-clusters`Operation mit den folgenden Parametern:

- **`--next-token`**—Optional. Wenn die Anzahl der ausgegebenen Elemente (`--max-results`) geringer als die Gesamtanzahl der Elemente ist, die von den zugrunde liegenden API-Aufrufen zurückgeliefert werden, enthält die Ausgabe ein `NextToken`. Dieses können Sie in einem anschließenden Befehl zum Abrufen der nächsten Gruppe von Elementen übergeben.
- **`--max-results`**—Optional. Die Gesamtzahl der Elemente, die in der Ausgabe des Befehls zurückgegeben werden sollen. Wenn mehr Ergebnisse als der angegebene `max-results` Wert vorhanden sind, wird ein Paginierungstoken (`next-token`) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.
  - Standard: 100
  - Mindestens 20, maximal 100

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic list-clusters \
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== \
    --max-results 2
```

Für Windows:

```
aws docdb-elastic list-clusters ^
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== ^
    --max-results 2
```

Die Ausgabe dieser Operation sieht etwa wie folgt aus:

```
{
   "Clusters": [
      {
         "ClusterIdentifier":"mycluster-1",
```

```
          "ClusterArn":"arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster"
          "Status":"available",
          "ClusterEndpoint":"sample-cluster.sharded-cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com"
       }
       {
          "ClusterIdentifier":"mycluster-2",
          "ClusterArn":"arn:aws:docdb:us-west-2:987654321098:sharded-cluster:sample-
cluster"
          "Status":"available",
          "ClusterEndpoint":"sample-cluster2.sharded-cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com"
       }
   ]
}
```

Löschen eines elastischen Clusters

In diesem Abschnitt wird erläutert, wie Sie einen elastischen Cluster mithilfe der AWS Management Console oder AWS CLI mit den folgenden Anweisungen löschen.

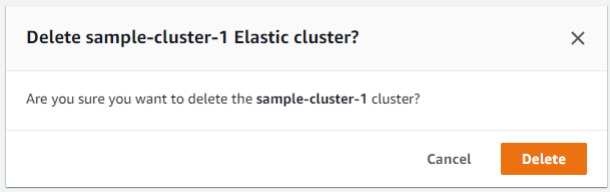
Using the AWS Management Console

So löschen Sie eine Elastic-Cluster-Konfiguration mit der AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Aktivieren Sie in der Clusterlistentabelle das Kontrollkästchen links neben dem Clusternamen, den Sie löschen möchten, und wählen Sie dann Aktionen aus. Wählen Sie im Dropdown-Menü Löschen aus.
4. Wählen Sie im Dialogfeld „Clustername“ Elastic Cluster löschen? die Option Löschen aus.



Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Status des Clusters finden Sie unter Überwachen des Status eines Amazon DocumentDB-Clusters.

Using the AWS CLI

Um einen elastischen Cluster mit der zu löschen AWS CLI, verwenden Sie die -`delete-cluster`Operation mit den folgenden Parametern:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, den Sie löschen möchten.
- **`--no-skip-final-backup`**—Optional. Wenn Sie eine endgültige Sicherung wünschen, müssen Sie diesen Parameter mit einem Namen für die endgültige Sicherung einschließen. Sie müssen entweder `--final-backup-identifier` oder `--skip-final-backup` angeben.
- **`--skip-final-backup`**—Optional. Verwenden Sie diesen Parameter nur, wenn Sie vor dem Löschen Ihres Clusters kein endgültiges Backup erstellen möchten. Standardmäßig wird ein letzter Snapshot erstellt.

In den folgenden AWS CLI Codebeispielen wird ein Cluster mit dem ARN arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster mit einem endgültigen Backup gelöscht.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic delete-cluster \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster \
    --no-skip-final-backup \
    --final-backup-identifier finalArnBU-arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

Für Windows:

```
aws docdb-elastic delete-cluster ^
```

```
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster ^
    --no-skip-final-backup ^
    --final-backup-identifier finalArnBU-arn:aws:docdb:us-
west-2:123456789012:sharded-cluster:sample-cluster
```

In den folgenden AWS CLI Codebeispielen wird ein Cluster mit dem ARN arn:aws:docdb:us-west- 2:123456789012:sharded-cluster:sample-cluster gelöscht, ohne ein endgültiges Backup zu erstellen.

Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic delete-cluster \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster \
    --skip-final-backup \
```

Für Windows:

```
aws docdb-elastic delete-cluster ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster ^
    --skip-final-backup ^
```

Die Ausgabe des `delete-cluster` Vorgangs ist eine Anzeige des Clusters, den Sie löschen.

Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zum Überwachen des Status des Clusters finden Sie unter Überwachen des Status eines Amazon DocumentDB-Clusters.

Verwalten von Elastic-Cluster-Snapshots

Manuelle Snapshots können erstellt werden, nachdem ein elastischer Cluster erstellt wurde. Automatisierte Backups werden erstellt, sobald der elastische Cluster-Snapshot erstellt wird.

> **Note**
>
> Ihr elastischer Cluster muss sich im `Available` Status befinden, damit ein manueller Snapshot erstellt werden kann.

In diesem Abschnitt wird erläutert, wie Sie Elastic-Cluster-Snapshots erstellen, anzeigen, wiederherstellen und löschen können.

Die folgenden Themen zeigen, wie Sie verschiedene Aufgaben ausführen, wenn Sie mit Amazon DocumentDB-Snapshots für elastische Cluster arbeiten.

Themen

Erstellen eines manuellen Elastic-Cluster-Snapshots

In diesem Abschnitt wird erläutert, wie Sie einen manuellen Snapshot eines elastischen Clusters mit der AWS Management Console oder AWS CLI mit den folgenden Anweisungen erstellen.

Using the AWS Management Console

So erstellen Sie einen manuellen Elastic-Cluster-Snapshot mit der AWS Management Console:

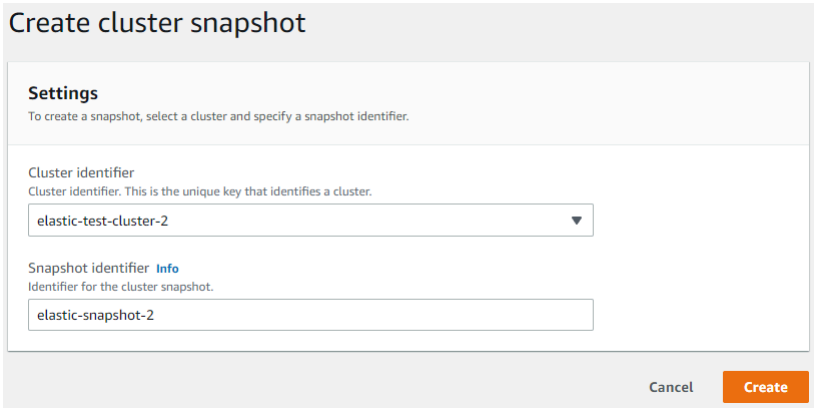
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie auf der Seite Snapshots die Option Create (Erstellen) aus.
4. Wählen Sie auf der Seite Cluster-Snapshot erstellen im Feld Cluster-ID Ihren elastischen Cluster aus der Dropdown-Liste aus.

   Geben Sie im Feld Snapshot-Kennung eine eindeutige Kennung für Ihren elastischen Cluster ein.

   Wählen Sie Erstellen.



> **Note**
>
> Alternativ können Sie auf das Dialogfeld Cluster-Snapshot erstellen zugreifen, indem Sie die Seite Cluster aufrufen, das Kontrollkästchen neben Ihrem Cluster aktivieren und dann Aktionen und dann Snapshot erstellen auswählen.

Ihr elastischer Cluster-Snapshot wird jetzt bereitgestellt. Dies kann bis zu einigen Minuten dauern. Sie können Ihren Snapshot anzeigen und wiederherstellen, wenn der Status als `Available` in der Liste Snapshots angezeigt wird.

Using the AWS CLI

Um einen manuellen Elastic-Cluster-Snapshot mit der zu erstellen AWS CLI, verwenden Sie die `-create-cluster-snapshot`Operation mit den folgenden Parametern:

- **`--snapshot-name`**—Erforderlich. Der Name des Cluster-Snapshots, den Sie erstellen möchten.
- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, für den Sie einen Snapshot erstellen möchten.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic create-cluster-snapshot \
    --snapshot-name sample-snapshot-1 \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster
```

Für Windows:

```
aws docdb-elastic create-cluster-snapshot ^
    --snapshot-name sample-snapshot-1 ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster
```

Anzeigen eines elastischen Cluster-Snapshots

In diesem Abschnitt wird erläutert, wie Sie Snapshot-Informationen für elastische Cluster mithilfe der AWS Management Console oder AWS CLI mit den folgenden Anweisungen anzeigen.

Using the AWS Management Console

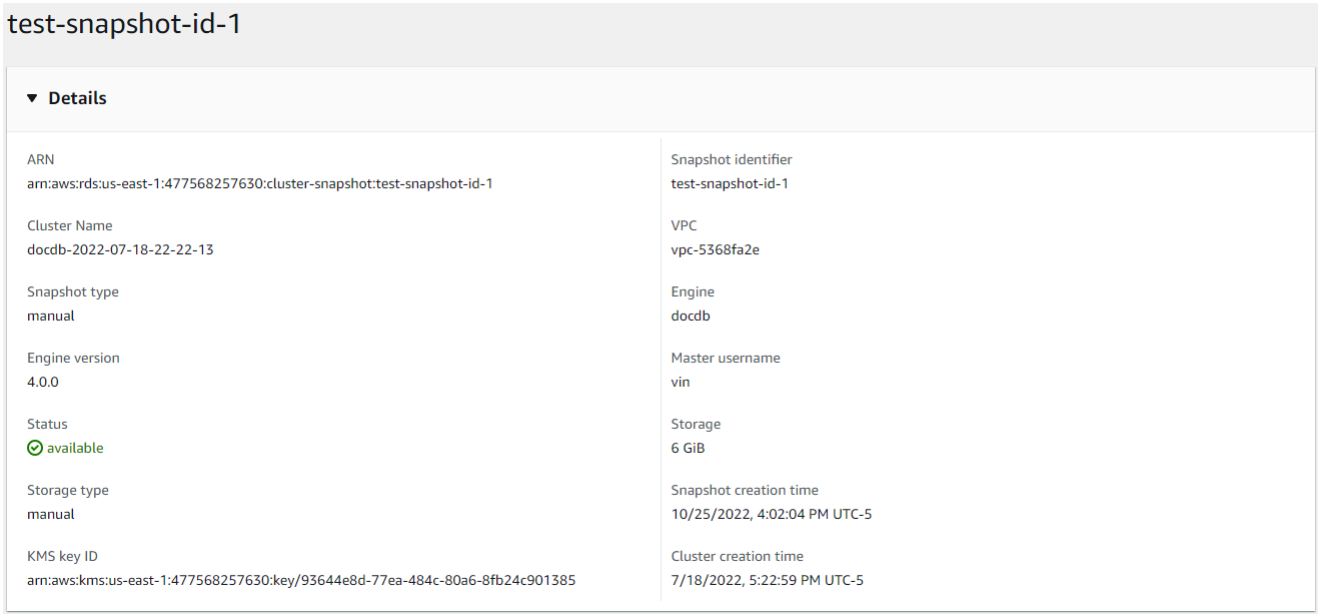
So zeigen Sie Informationen zu einem bestimmten Elastic-Cluster-Snapshot mithilfe der an AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.

2. Wählen Sie im Navigationsbereich die Option Snapshots.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie auf der Seite Snapshots Ihren Snapshot aus der Liste aus, indem Sie auf den Namen in der Spalte Snapshot-Kennung klicken.
4. Zeigen Sie die Informationen Ihres Snapshots unter Details an.



Using the AWS CLI

Um Informationen zu einem bestimmten Elastic-Cluster-Snapshot mithilfe der anzuzeigen AWS CLI, verwenden Sie die -`get-cluster-snapshot`Operation mit den folgenden Parametern:

- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, für den Sie Informationen benötigen.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Für Windows:

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Um Informationen zu einem bestimmten Elastic-Cluster-Snapshot mithilfe der anzuzeigen AWS CLI, verwenden Sie die -`get-cluster-snapshot`Operation mit den folgenden Parametern:

- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, für den Sie Informationen benötigen.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Für Windows:

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Um Informationen zu allen elastischen Cluster-Snapshots mithilfe der anzuzeigen AWS CLI, verwenden Sie die -`list-cluster-snapshots`Operation mit den folgenden Parametern:

- **`--snapshot-type`**—Optional. Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:
  - `automated` – Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr AWS Konto erstellt hat.
  - `manual` – Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr AWS Konto erstellt haben.

  - `shared` – Gibt alle manuellen Cluster-Snapshots zurück, die für Ihr AWS Konto freigegeben wurden.
  - `public` – Gibt alle Cluster-Snapshots zurück, die als öffentlich markiert wurden.
- **`--next-token`**—Optional. Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn dieser Parameter angegeben ist, enthält die Antwort nur Datensätze über dieses Token hinaus bis zu dem von angegebenen Wert`max-results`.
- **`--max-results`**—Optional. Die maximale Anzahl der Ergebnisse, die in die Antwort aufgenommen werden sollen. Wenn mehr Ergebnisse als der angegebene `max-results` Wert vorhanden sind, wird ein Paginierungstoken (`next-token`) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.
  - Standard: 100
  - Mindestens 20, maximal 100

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic list-cluster-snapshots \
    --snapshot-type  value \
    --next-token  value \
    --max-results 50
```

Für Windows:

```
aws docdb-elastic list-cluster-snapshots ^
    --snapshot-type  value ^
    --next-token  value ^
    --max-results 50
```

Wiederherstellen eines elastischen Clusters aus einem Snapshot

In diesem Abschnitt wird erläutert, wie Sie einen elastischen Cluster aus einem Snapshot mithilfe der AWS Management Console oder AWS CLI mit den folgenden Anweisungen wiederherstellen.

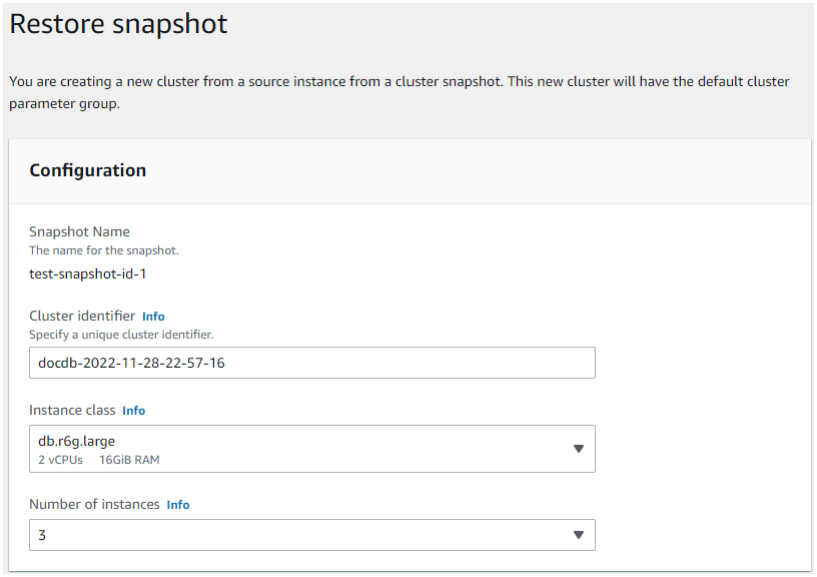
Using the AWS Management Console

So stellen Sie einen elastischen Cluster aus einem Snapshot mithilfe der wieder her AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Snapshot-ID die Schaltfläche links neben dem Snapshot aus, die Sie zum Wiederherstellen eines Clusters verwenden möchten.
4. Wählen Sie Aktionen und dann Wiederherstellen aus.



5. Geben Sie auf der Seite Snapshot wiederherstellen einen Namen für den neuen Cluster in das Feld Cluster-ID ein.

> **Note**
>
> Für jede manuelle Snapshot-Wiederherstellung müssen Sie einen neuen Cluster erstellen.

6. Wählen Sie im Feld Virtual Private Cloud (VPC) eine VPC aus der Dropdown-Liste aus.
7. Für Subnetze und VPC-Sicherheitsgruppen können Sie die Standardeinstellungen verwenden oder drei Subnetze Ihrer Wahl und bis zu drei VPC-Sicherheitsgruppen (mindestens eine) auswählen.
8. Wenn Sie mit der Cluster-Konfiguration zufrieden sind, wählen Sie Restore cluster (Cluster wiederherstellen) aus und warten Sie, bis Ihr Cluster wiederhergestellt ist.

Using the AWS CLI

Um einen elastischen Cluster mithilfe der aus einem Snapshot wiederherzustellen AWS CLI, verwenden Sie die `-restore-cluster-from-snapshot`Operation mit den folgenden Parametern:

- **`--cluster-name`**—Erforderlich. Der aktuelle Name des elastischen Clusters, wie er während der Erstellung oder letzten Änderung eingegeben wurde.
- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, der zur Wiederherstellung des Clusters verwendet wird.
- **`--vpc-security-group-ids`**—Optional. Eine oder mehrere Amazon EC2- und Amazon Virtual Private Cloud (VPC)-Sicherheitsgruppen, die dem Cluster zugeordnet werden sollen.
- **`--kms-key-id`**—Optional. Konfigurieren Sie die KMS-Schlüsselkennung für einen verschlüsselten Cluster.

  Die KMS-Schlüsselkennung ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon Web Services-Konto erstellen, das den KMS-Verschlüsselungsschlüssel besitzt, der zur Verschlüsselung des neuen Clusters verwendet wird, können Sie den KMS-Schlüssel-Alias anstelle des ARN für den KMS-Verschlüsselungsschlüssel verwenden.

  Wenn in kein Verschlüsselungsschlüssel angegeben ist KmsKeyId und der `StorageEncrypted` Parameter „true“ ist, verwendet Amazon DocumentDB Ihren Standardverschlüsselungsschlüssel.

- **`--subnet-ids`**—Optional. Netzwerk-Subnetz-IDs.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic restore-cluster-from-snapshot \
     --cluster-name elastic-sample-cluster \
     --snapshot-arn sampleResourceName \
     --vpc-security-group-ids value ec-65f40350 \
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Für Windows:

```
aws docdb-elastic restore-cluster-from-snapshot ^
     --cluster-name elastic-sample-cluster ^
     --snapshot-arn sampleResourceName ^
     --vpc-security-group-ids value ec-65f40350 ^
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Kopieren eines Elastic-Cluster-Snapshots

In Amazon DocumentDB können Sie manuelle und automatische Elastic-Cluster-Snapshots innerhalb derselben Region und innerhalb desselben Kontos kopieren. In diesem Abschnitt wird erläutert, wie Sie einen elastischen Cluster-Snapshot mit der AWS Management Console oder kopieren AWS CLI.

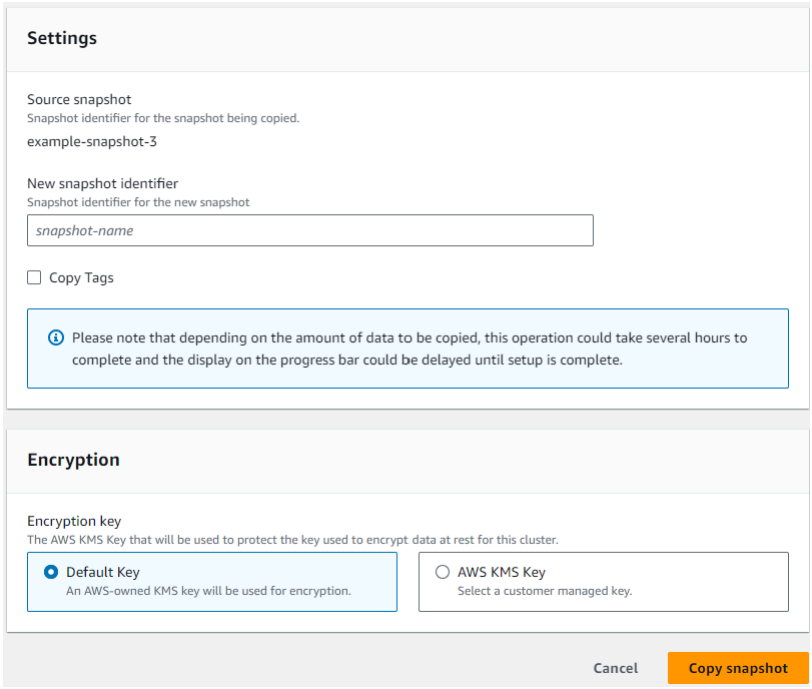
Using the AWS Management Console

So kopieren Sie einen elastischen Cluster-Snapshot mit der AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Snapshot-ID die Schaltfläche links neben dem Snapshot aus, den Sie kopieren möchten.
4. Wählen Sie Aktionen und dann Kopieren aus.



5. Geben Sie für Neue Snapshot-ID den Namen des neuen Snapshots ein.
6. Aktivieren Sie für Tags kopieren das Kontrollkästchen, wenn Sie alle Tags aus dem elastischen Quell-Cluster-Snapshot in den elastischen Ziel-Cluster-Snapshot kopieren möchten.
7. Wählen Sie für Verschlüsselung entweder einen Standard AWS -KMS-Schlüssel oder einen KMS-Schlüssel Ihrer Wahl aus. Mit der zweiten Option können Sie einen vorhandenen KMS-Schlüssel auswählen, den Sie bereits erstellt haben, oder einen neuen Schlüssel erstellen.
8. Wählen Sie Snapshot kopieren, wenn Sie fertig sind.

Using the AWS CLI

Um einen elastischen Cluster-Snapshot mit der zu kopieren AWS CLI, verwenden Sie die `-copy-cluster-snapshot`Operation mit den folgenden Parametern:

- **`--source-db-cluster-snapshot-identifier`**—Erforderlich. Die Kennung des vorhandenen Elastic-Cluster-Snapshots, der kopiert wird. Der elastische Cluster-Snapshot muss vorhanden sein und sich im Status „Verfügbar“ befinden. Wenn Sie den Snapshot in eine andere kopieren AWS-Region, muss diese Kennung im ARN-Format für die Quell- vorliegen AWS-Region. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- **`--target-db-cluster-snapshot-identifier`**—Erforderlich. Die Kennung des neuen elastischen Cluster-Snapshots, der aus dem vorhandenen Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

  Einschränkungen des Ziel-Snapshot-Namens:

  - Kann nicht der Namen eines vorhandenen Snapshots sein.
  - Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic copy-cluster-snapshot \
     --source-cluster-snapshot-arn <sample ARN> \
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Für Windows:

```
aws docdb-elastic copy-cluster-snapshot ^
     --source-cluster-snapshot-arn <sample ARN> ^
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Löschen eines elastischen Cluster-Snapshots

In diesem Abschnitt wird erläutert, wie Sie einen elastischen Cluster-Snapshot mithilfe von AWS Management Console oder löschen AWS CLI.

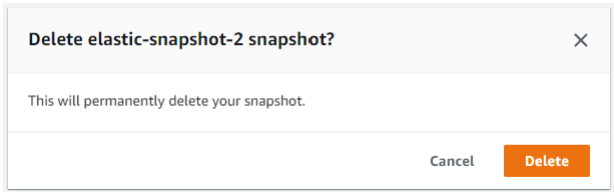
Using the AWS Management Console

So stellen Sie einen elastischen Cluster aus einem Snapshot mithilfe der wieder her AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Snapshot-ID die Schaltfläche links neben dem Snapshot aus, die Sie zum Wiederherstellen eines Clusters verwenden möchten.
4. Wählen Sie Aktionen und dann Löschen aus.



5. Wählen Sie im Dialogfeld Snapshot löschen die Option Löschen aus.

Using the AWS CLI

Um einen elastischen Cluster-Snapshot mit der zu löschen AWS CLI, verwenden Sie die `-delete-cluster-snapshot`Operation mit den folgenden Parametern:

- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, der zur Wiederherstellung des Clusters verwendet wird.

Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic delete-cluster-snapshot \
     --snapshot-arn sampleResourceName
```

Für Windows:

```
aws docdb-elastic delete-cluster-snapshot ^
     --snapshot-arn sampleResourceName
```

Verwalten eines automatischen Backups für Elastic-Cluster-Snapshots

Amazon DocumentDB erstellt tägliche Snapshots Ihrer elastischen Cluster. Sie können das bevorzugte Backup-Fenster und den Aufbewahrungszeitraum für Backups in einer neuen oder vorhandenen Konfiguration für elastische Cluster-Snapshots angeben. In diesem Abschnitt wird erläutert, wie Sie automatische Backup-Parameter in einem elastischen Cluster-Snapshot festlegen, indem Sie entweder die AWS Management Console oder verwenden AWS CLI.

Using the AWS Management Console

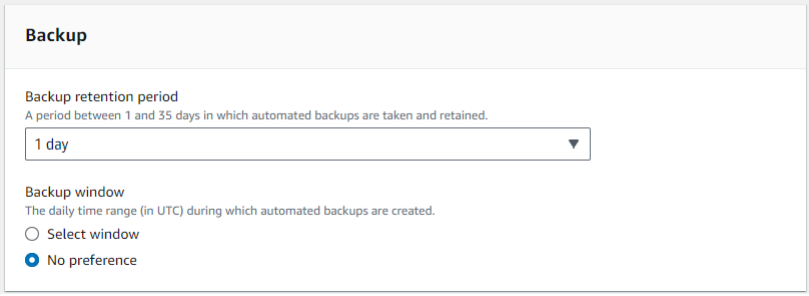
So legen Sie ein automatisches Backup für einen neuen Elastic-Cluster-Snapshot mithilfe der fest AWS Management Console:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite Ihres Bildschirms nicht angezeigt wird, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Cluster-ID die Schaltfläche links neben dem Cluster aus, für den Sie die Backup-Einstellungen ändern möchten.

4. Wählen Sie Aktionen und dann Ändern aus.
5. Bearbeiten Sie im Abschnitt Backup die Felder entsprechend Ihren Backup-Anforderungen.



   a. Aufbewahrungszeitraum für Backups – Wählen Sie in der Liste die Anzahl der Tage aus, für die automatische Backups dieses Clusters aufbewahrt werden sollen, bevor sie gelöscht werden.
   b. Backup-Fenster – Legen Sie die tägliche Zeit und Dauer fest, in der Amazon DocumentDB Backups dieses Clusters erstellen soll.

      i. Wählen Sie Fenster auswählen, wenn Sie die Zeit und Dauer der Erstellung von Backups konfigurieren möchten.

         Startzeit – Wählen Sie in der ersten Liste die Startzeitstunde (UTC) zum Starten Ihrer automatischen Backups aus. Wählen Sie in der zweiten Liste die Minute für den Beginn der automatischen Backups aus.

         Dauer – Wählen Sie in der Liste die Anzahl der Stunden aus, die dem Erstellen automatischer Backups zugewiesen werden sollen.

      ii. Wählen Sie Keine Präferenz, wenn Amazon DocumentDB den Zeitpunkt und die Dauer der Erstellung von Backups auswählen soll.
6. Wählen Sie Cluster ändern, wenn Sie fertig sind.

Using the AWS CLI

Um eine automatische Sicherung für einen neuen elastischen Cluster-Snapshot mithilfe der einzurichten AWS CLI, verwenden Sie die `-create-cluster-snapshot`Operation mit den folgenden Parametern:

- **`--preferred-backup-window`**—Optional. Der täglich bevorzugte Zeitraum, in dem automatische Backups erstellt werden. Die Standardeinstellung ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock für jede ausgewählt wird AWS-Region.

Einschränkungen:

- Muss im Format `hh24:mi-hh24:mi` angegeben werden.
- Muss in Universal Coordinated Time (UTC) angegeben werden.
- Darf nicht mit dem bevorzugten Wartungsfenster in Konflikt treten.
- Muss mindestens 30 Minuten betragen.

- **`--backup-retention-period`**—Optional. Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Der Standardwert lautet 1.

  Einschränkungen:

  - Muss einen Mindestwert von 1 angeben.
  - Der Bereich liegt zwischen 1 und 35.

> **Note**
>
> Automatisierte Backups werden nur erstellt, wenn sich der Cluster im Status „aktiv“ befindet.

> **Note**
>
> Sie können die `backup-retention-period` Parameter `preferred-backup-window` und eines vorhandenen elastischen Clusters auch mit dem `aws docdb-elastic update-cluster` Befehl ändern.

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Im folgenden `create-cluster` Beispiel wird der Amazon DocumentDB*`-Beispiel-Cluster`* mit dem Aufbewahrungszeitraum für automatische Backups von *`7`* Tagen und einem bevorzugten Backup-Fenster von *`18:00 bis 18:30 UTC`* erstellt.

Für Linux, macOS oder Unix:

```
aws docdb-elastic create-cluster \
```

```
    --cluster-name sample-cluster \
    --shard-capacity 2 \
    --shard-count 2 \
    --admin-user-name SampleAdmin \
    --auth-type PLAIN_TEXT \
    --admin-user-password SamplePass123! \
    --preferred-backup-window 18:00-18:30 \
    --backup-retention-period 7
```

Für Windows:

```
aws docdb-elastic create-cluster ^
    --cluster-name sample-cluster ^
    --shard-capacity 2 ^
    --shard-count 2 ^
    --admin-user-name SampleAdmin ^
    --auth-type PLAIN_TEXT ^
    --admin-user-password SamplePass123! ^
    --preferred-backup-window 18:00-18:30 ^
    --backup-retention-period 7
```

Stoppen und Starten eines elastischen Amazon DocumentDB-Clusters

Das Stoppen und Starten elastischer Amazon DocumentDB-Cluster kann Ihnen helfen, die Kosten für Entwicklungs- und Testumgebungen zu verwalten. Anstatt bei jeder Verwendung von Amazon DocumentDB elastische Cluster zu erstellen und zu löschen, können Sie Ihren Cluster vorübergehend stoppen, wenn er nicht benötigt wird. Sie können es dann erneut starten, wenn Sie Ihre Tests fortsetzen.

Themen

- Übersicht über das Stoppen und Starten eines elastischen Clusters
- Operationen, die Sie auf einem gestoppten elastischen Cluster ausführen können

Übersicht über das Stoppen und Starten eines elastischen Clusters

In Zeiten, in denen Sie keinen elastischen Amazon DocumentDB-Cluster benötigen, können Sie den Cluster stoppen. Bei Bedarf können Sie den Cluster jederzeit erneut starten. Das Starten und Stoppen vereinfacht die Einrichtungs- und Außerbetriebnahmeprozesse für Elastic Cluster,

die für Entwicklungs-, Test- oder ähnliche Aktivitäten verwendet werden, die keine kontinuierliche Verfügbarkeit erfordern. Sie können einen elastischen Cluster mit der AWS Management Console oder der AWS CLI mit einer einzigen Aktion anhalten und starten.

Während Ihr elastischer Cluster gestoppt ist, bleibt das Cluster-Speichervolumen unverändert. Sie zahlen nur für Speicherung, manuelle Snapshots und automatischen Sicherungsspeicher innerhalb des angegebenen Aufbewahrungsfensters. Amazon DocumentDB startet Ihren elastischen Cluster nach sieben Tagen automatisch, damit er nicht hinter erforderlichen Wartungsupdates zurückbleibt. Wenn Ihr Cluster nach sieben Tagen beginnt, wird Ihnen die Verwendung des elastischen Clusters wieder in Rechnung gestellt. Während Ihr Cluster gestoppt ist, können Sie Ihr Speichervolume nicht abfragen, da die Abfrage erfordert, dass sich der Cluster im Status verfügbar befindet.

Wenn ein elastischer Amazon DocumentDB-Cluster gestoppt wird, kann der Cluster in keiner Weise geändert werden. Dazu gehört auch das Löschen des Clusters.

Using the AWS Management Console

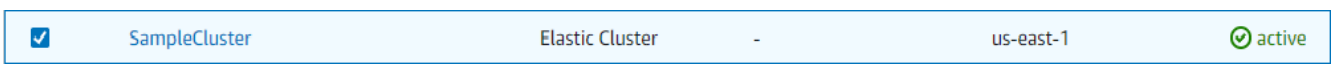
Das folgende Verfahren zeigt Ihnen, wie Sie einen elastischen Cluster im verfügbaren Zustand stoppen oder einen gestoppten elastischen Cluster starten.

So stoppen oder starten Sie einen elastischen Amazon DocumentDB-Cluster

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

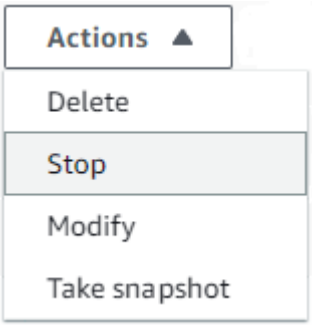
   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Namen des Clusters aus, den Sie stoppen oder starten möchten.

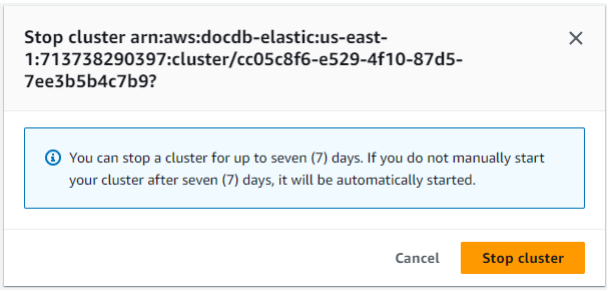
   

4. Wählen Sie Aktionen aus und dann die Aktion, die Sie auf dem Cluster ausführen möchten.
   - Wenn Sie den Cluster stoppen möchten und der Cluster verfügbar ist:

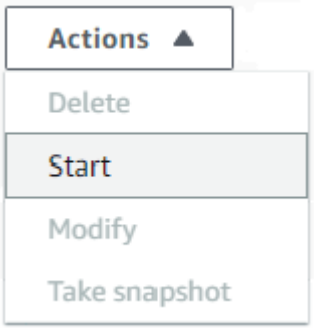
a. Wählen Sie Beenden aus.



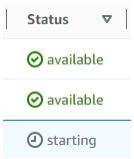
b. Bestätigen Sie im Bestätigungsdialogfeld, dass Sie den elastischen Cluster stoppen möchten, indem Sie Cluster stoppen auswählen, oder wählen Sie Abbrechen, um den Cluster laufen zu lassen.



- Wenn Sie einen gestoppten Cluster starten möchten, wählen Sie Starten aus.



5. Überwachen Sie den Status des elastischen Clusters. Wenn Sie den Cluster gestartet haben, können Sie den Cluster wieder verwenden, wenn der Cluster verfügbar ist. Weitere Informationen finden Sie unter Ermitteln des Cluster-Status.



Using the AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie einen elastischen Cluster im aktiven oder verfügbaren Zustand stoppen oder einen gestoppten elastischen Cluster starten.

Um einen elastischen Cluster mit der zu stoppen AWS CLI, verwenden Sie die `-stop-cluster`Operation. Verwenden Sie die `start-cluster`-Operation, um einen gestoppten Cluster zu starten. Beide Operationen verwenden den `--cluster-arn`-Parameter.

Parameter:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des elastischen Clusters, den Sie anhalten oder starten möchten.

Example – So halten Sie einen elastischen Cluster mit der an AWS CLI

Ersetzen Sie im folgenden Beispiel jeden *`Platzhalter für Benutzereingaben`* durch Ihre eigenen Informationen.

Der folgende Code stoppt den elastischen Cluster mit dem ARN `arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`.

> **Note**
>
> Der elastische Cluster muss sich im aktiven oder verfügbaren Zustand befinden.

Für Linux, macOS oder Unix:

```
aws docdb-elastic stop-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Für Windows:

```
aws docdb-elastic stop-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Example – So starten Sie einen elastischen Cluster mit der AWS CLI

Ersetzen Sie im folgenden Beispiel jeden *Platzhalter für Benutzereingaben* durch Ihre eigenen Informationen.

Der folgende Code startet den elastischen Cluster mit dem ARN `arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`.

> **Note**
>
> Der elastische Cluster muss derzeit gestoppt werden.

Für Linux, macOS oder Unix:

```
aws docdb-elastic start-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Für Windows:

```
aws docdb-elastic start-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Operationen, die Sie auf einem gestoppten elastischen Cluster ausführen können

Sie können die Konfiguration eines elastischen Amazon DocumentDB-Clusters nicht ändern, während der Cluster gestoppt ist. Vor solchen administrativen Aktionen müssen Sie den Cluster starten.

Amazon DocumentDB wendet alle geplanten Wartungsarbeiten auf Ihren angehaltenen elastischen Cluster erst an, nachdem er erneut gestartet wurde. Nach sieben Tagen startet Amazon DocumentDB automatisch einen gestoppten elastischen Cluster, sodass er in Bezug auf seinen Wartungsstatus nicht zu weit zurückfällt. Wenn der elastische Cluster neu gestartet wird, werden Ihnen die Shards im Cluster erneut in Rechnung gestellt.

Während ein elastischer Cluster gestoppt wird, führt Amazon DocumentDB keine automatisierten Backups durch und verlängert auch nicht den Aufbewahrungszeitraum für Backups.

Datenverschlüsselung im Ruhezustand im Ruhezustand und Deaktivieren Datenverschlüsselung Amazon DocumentDB DB-Cluster

Die folgenden Themen helfen Ihnen dabei, sich mit AWS Key Management Service Verschlüsselungsschlüsseln für Amazon DocumentDB DocumentDB-Elastic-Cluster vertraut zu machen, diese zu erstellen und zu überwachen:

Themen

Amazon DocumentDB Elastic Clusters lassen sich für die Schlüsselverwaltung automatisch in AWS Key Management Service (AWS KMS) integrieren und verwenden zum Schutz Ihrer Daten eine Methode, die als Umschlagverschlüsselung bezeichnet wird. Weitere Informationen zur Envelope-Verschlüsselung finden Sie unter Envelope-Verschlüsselung im AWS Key Management Service-Entwicklerhandbuch.

Importieren in &S3;AWS KMS keyDies ist eine logische Darstellung eines Schlüssels. Der KMS-Schlüssel enthält Metadaten wie die Schlüssel-ID, das Erstellungsdatum, die Beschreibung und den Schlüsselstatus. Der KMS-Schlüssel enthält auch das zur Ver- und Entschlüsselung von Daten verwendete Schlüsselmaterial. Weitere Informationen über KMS-Schlüssel finden Sie unter AWS KMS keys im AWS Key Management Service Developer Guide.

Elastic Cluster von Amazon DocumentDB unterstützen Verschlüsselung mit zwei Arten von Schlüsseln:

- AWSeigene Schlüssel — Amazon DocumentDB Elastic Clusters verwenden diese Schlüssel standardmäßig, um personenbezogene Daten automatisch zu verschlüsseln. Sie können keine Schlüssel, verwalten und verwalten, verwenden und verwenden, verwalten und verwenden, verwalten und verwenden, AWS verwalten und verwenden, verwenden und verwalten, verwenden und verwalten, verwenden und verwenden, Sie müssen jedoch keine Maßnahmen

oder Programme ändern, um die Schlüssel zur Datenverschlüsselung und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Weitere Informationen finden Sie unter AWSOwned Keys im AWS Key Management ServiceDeveloper Guide.

- KundenverwalteteAWS KMS keys, besitzen und verwalten. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie folgende Aufgaben ausführen:
  - Festlegung und Aufrechterhaltung wichtiger Richtlinien
  - Festlegung und Pflege von IAM-Richtlinien und Zuschüssen
  - Aktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren und Deaktivieren
  - Drehbares Schlüsselverschlüsseltes kryptografisches Schlüsselmaterial
  - Hinzufügen von Tags
  - Schlüsselaliase erstellen
  - Schlüssel für das Löschen planen

  Weitere Informationen finden Sie im AWS Key Management ServiceEntwicklerhandbuch unter Vom Kunden verwaltete Schlüssel.

> **Important**
>
> Sie müssen einen symmetrischen Verschlüsselungs-KMS verwenden, um Ihren Cluster zu verschlüsseln, da Amazon DocumentDB nur symmetrische Verschlüsselungs-KMS-Schlüssel unterstützt. Verwenden Sie keinen asymmetrischen KMS-Schlüssel, um die Daten in Ihren Amazon DocumentDB Elastic Clusters zu verschlüsseln. Weitere Informationen finden Sie AWS KMSim AWS Key Management ServiceEntwicklerhandbuch unter Asymmetrische Schlüssel.
>
> Wenn Amazon DocumentDB keinen Zugriff mehr auf den Verschlüsselungsschlüssel für einen Cluster erhalten kann - zum Beispiel, wenn der Zugriff auf einen Schlüssel widerrufen wird - geht der verschlüsselte Cluster in einen Endzustand über. In diesem Fall können Sie den Cluster nur aus einer Sicherung wiederherstellen. Für Amazon DocumentDB sind Backups immer für einen Tag aktiviert. Wenn Sie den Schlüssel für einen verschlüsselten Amazon DocumentDB-Cluster deaktivieren, verlieren Sie außerdem irgendwann den Lese- und Schreibzugriff auf diesen Cluster. Wenn Amazon DocumentDB auf einen Cluster trifft,

der durch einen Schlüssel verschlüsselt ist, auf den es keinen Zugriff hat, versetzt es den Cluster in einen Endzustand. In diesem Fall ist der Cluster nicht länger verfügbar und der aktuelle Zustand der Datenbank kann nicht mehr wiederhergestellt werden. Um den Cluster wiederherzustellen, müssen Sie den Zugriff auf den Verschlüsselungsschlüssel für Amazon DocumentDB erneut aktivieren und den Cluster anschließend aus einer Sicherungsdatei wiederherstellen.

Important

Sie können den KMS-Schlüssel für einen verschlüsselten Cluster nicht mehr ändern, nachdem Sie ihn bereits erstellt haben. Stellen Sie sicher, die Anforderungen für Ihren Verschlüsselungsschlüssel zu definieren, bevor Sie Ihr verschlüsseltes Elastic Cluster erstellen.

So verwenden Elastic-Cluster von Amazon DocumentDB Zuschüsse in AWS KMS

Amazon DocumentDB Elastic Clusters benötigen eine Genehmigung, um Ihren kundenverwalteten Schlüssel verwenden zu können.

Wenn Sie einen Cluster erstellen, die/der mit einem kundenverwalteten Schlüssel verschlüsselt ist, erstellen Amazon DocumentDB Elastic Clusters in Ihrem Namen eine Genehmigung, indem sie eine `CreateGrant` Anfrage an AWS KMS senden. Genehmigungen in AWS KMS werden verwendet, um Amazon DocumentDB Elastic Clusters Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

Amazon DocumentDB Elastic Clusters benötigen die Genehmigung, Ihren Kunden verwalteten Schlüssel für die folgenden internen Vorgänge zu verwenden:

- Senden Sie `DescribeKey` Anfragen an, AWS KMS um zu überprüfen, ob die symmetrische, vom Kunden verwaltete KMS-Schlüssel-ID, die beim Erstellen einer Tracker- oder Geofence-Sammlung eingegeben wurde, gültig ist.
- Senden Sie `GenerateDataKey` Anfragen an, AWS KMS um Datenschlüssel zu generieren, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden.

- Senden Sie `Decrypt` Anfragen AWS KMS an, die verschlüsselten Datenschlüssel zu entschlüsseln, damit Sie diese zur Verschlüsselung Ihrer Daten verwenden können.
- Sie können den Zugriff auf die Genehmigung jederzeit widerrufen oder den Zugriff des Services auf den vom Kunden verwalteten Schlüssel entfernen. Wenn Sie dies tun, können Amazon DocumentDB Elastic Clusters nicht auf die vom kundenverwalteten Schlüssel verschlüsselten Daten zugreifen, was sich auf Vorgänge auswirkt, die von diesen Daten abhängig sind.

Erstellen und verwalten verwalten verwalten verwalten verwalten verwalten verwalten verwalten verwalten

Sie können einen symmetrischen, kundenverwalteten Schlüssel erstellen, indem Sie die AWS Management Console oder die AWS KMS -API verwenden.

Symmetrische kundenverwaltete Schlüsselerstellung

Folgen Sie den Schritten zur Erstellung eines symmetrischen, vom Kunden verwalteten Schlüssels im AWS Key Management ServiceEntwicklerhandbuch.

Schlüsselrichtlinie

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie in den Informationen zum KMS-Schlüsselzugriff in der AWS Key Management ServiceÜbersicht des AWS Key Management ServiceEntwicklerhandbuchs.

Um Ihren kundenverwalteten Schlüssel mit den Elastic-Cluster-Ressourcen von Amazon DocumentDB verwenden zu können, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zulässig sein:

- `kms:CreateGrant`— Fügt einem vom Kunden verwalteten Schlüssel einen Zuschuss hinzu. Gewährt Kontrollzugriff auf einen bestimmten KMS-Schlüssel, der den Zugriff auf die von Amazon Location Service benötigten Vorgänge ermöglicht. Weitere Informationen zur Verwendung von Zuschüssen finden Sie AWS KMSim AWS Key Management ServiceEntwicklerhandbuch unter Zuschüsse.
- `kms:DescribeKey`— Stellt die kundenverwalteten Schlüsseldetails bereit, damit Docdb Elastic den Schlüssel validieren kann.

- `kms:Decrypt`— Ermöglicht Docdb Elastic, den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf verschlüsselte Daten zu verwenden.
- `kms:GenerateDataKey`— Ermöglicht Docdb Elastic, einen verschlüsselten Datenschlüssel zu generieren und zu speichern, da der Datenschlüssel nicht sofort zum Verschlüsseln verwendet wird.

Weitere Informationen finden Sie unter Berechtigungen für AWS Dienste in den wichtigsten Richtlinien und unter Problembehandlung bei Schlüsselzugriffen im AWS Key Management ServiceEntwicklerhandbuch.

Beschränkung des vom Kunden verwalteten Schlüsselzugriffs über IAM-Richtlinien

Zusätzlich zu den KMS-Schlüsselrichtlinien können Sie in einer IAM-Richtlinie auch die KMS-Schlüsselberechtigungen einschränken.

Sie können die IAM-Richtlinie auf verschiedene Weise strikter gestalten. Damit der kundenverwaltete Schlüssel beispielsweise nur für Anforderungen aus Amazon DocumentDB Elastic Clusters verwendet werden kann, können Sie den `kms:ViaService`Bedingungsschlüssel mit dem `docdb-elastic.<region-name>.amazonaws.com` Wert verwenden.

Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung eines KMS-Schlüssels erlauben im AWS Key Management Service-Entwicklerhandbuch.

Überwachen und Deaktivieren Ihre Verschlüsselungsschlüssel für Amazon DocumentDB Elastic Clusters überwachen

Wenn Sie einen vom AWS KMS key Kunden verwalteten Schlüssel mit Ihren Docdb Elastic-Ressourcen verwenden, können Sie Amazon CloudWatch Logs verwendenAWS CloudTrail, um Anfragen zu verfolgen, an die Docdb Elastic sendet. AWS KMS

Die folgenden Beispiele sind AWS CloudTrail Ereignisse für`CreateGrant`,, und `DescribeKey` zur Überwachung von AWS KMS key Vorgängen `GenerateDataKeyWithoutPlainTextDecrypt`, die von Elastic-Clustern von Amazon DocumentDB aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

CreateGrant

```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-09T23:04:20Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-09T23:55:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "retiringPrincipal": "docdb-elastic.us-east-1.amazonaws.com",
        "granteePrincipal": "docdb-elastic.us-east-1.amazonaws.com",
        "operations": [
            "Decrypt",
            "Encrypt",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "CreateGrant",
            "RetireGrant",
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
```

```
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKey

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
```

```
            "attributes": {
                "creationDate": "2023-05-10T18:02:59Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-10T18:03:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Decrypt


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-10T18:05:49Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-10T18:06:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
```

```
}
```

DescribeKey

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-09T23:04:20Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-09T23:55:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "keyId": "alias/SampleKmsKey"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
```

```
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Weitere Informationen

Die folgenden Ressourcen enthalten weitere Informationen zur Datenverschlüsselung im Ruhezustand:

- Weitere Informationen zu AWS KMS Konzepten finden Sie unter AWS Key Management ServiceGrundkonzepte im AWS Key Management ServiceEntwicklerhandbuch.
- Weitere Informationen zur AWS KMS Sicherheit finden Sie unter Bewährte Sicherheitsmethoden AWS Key Management Service im AWS Key Management ServiceEntwicklerhandbuch.

Serviceverknüpfte Rollen in elastischen Clustern

Elastische Amazon DocumentDB-Cluster verwenden AWS Identity and Access Management (IAM) serviceverknüpfte Rollen . Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit elastischen Amazon DocumentDB-Clustern verknüpft ist. Serviceverknüpfte Rollen werden von elastischen Amazon DocumentDB-Clustern vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer -AWSServices in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Verwendung elastischer Amazon DocumentDB-Cluster, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Elastische Amazon DocumentDB-Cluster definieren die Berechtigungen ihrer serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, können nur elastische Amazon DocumentDB-Cluster ihre Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden. Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies

schützt Ihre elastischen Amazon DocumentDB-Cluster-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter -AWSServices, die mit IAM funktionieren. Suchen Sie nach den Services, die in der Spalte Serviceverknüpfte Rolle mit Ja markiert sind. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für Elastic Cluster

Elastische Amazon DocumentDB-Cluster verwenden die serviceverknüpfte Rolle namens `AWSServiceRoleForDocDB-Elastic`, um elastischen Amazon DocumentDB-Clustern den Aufruf von -AWSServices im Namen Ihrer Cluster zu ermöglichen.

Dieser dienstgebundenen Rolle ist eine Berechtigungsrichtlinie namens `AmazonDocDB-ElasticServiceRolePolicy` zugeordnet, die ihr Berechtigungen für den Betrieb in Ihrem Konto erteilt. Die Rollenberechtigungsrichtlinie erlaubt es elastischen Amazon DocumentDB-Clustern, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Wenn die folgende Fehlermeldung angezeigt wird: „Die Ressource konnte nicht erstellt werden. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später erneut.“ Stellen Sie sicher, dass Sie die folgenden Berechtigungen aktiviert haben:

```
{
"Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-elastic.amazonaws.com/
AWSServiceRoleForDocDB-Elastic",
    "Condition": {
"StringLike": {
"iam:AWSServiceName":"docdb-elastic.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter Berechtigungen von serviceverknüpften Rollen im AWS Benutzerhandbuch für Identity and Access Management.

Erstellen einer serviceverknüpften Rolle für elastische Amazon DocumentDB-Cluster

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine DB-Instance erstellen, erstellt Amazon DocumentDB Elastic Cluster die serviceverknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für elastische Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster erlauben es Ihnen nicht, die `AWSServiceRoleForDocDB-Elastic` serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im AWS Benutzerhandbuch für Identity and Access Management.

Löschen einer serviceverknüpften Rolle für elastische Amazon DocumentDB-Cluster

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch alle Ihre -Cluster löschen, bevor Sie die serviceverknüpfte Rolle löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie, ob die serviceverknüpfte Rolle über eine aktive Sitzung in der IAM-Konsole verfügt:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole.
2. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus. Wählen Sie dann den Namen (nicht das Kontrollkästchen) der Rolle `AWSServiceRoleForDocDB-Elastic` aus.
3. Wählen Sie auf der Seite Summary (Zusammenfassung) für die ausgewählte Rolle die Registerkarte Access Advisor (Advisor aufrufen) aus.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob elastische Amazon DocumentDB-Cluster die `AWSServiceRoleForDocDB-Elastic` Rolle verwenden, können Sie versuchen, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt das Löschen fehl und Sie können die anzeigenAWS-Regionen, in der die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.
> Wenn Sie die `AWSServiceRoleForDocDB-Elastic` Rolle entfernen möchten, müssen Sie zunächst alle Ihre Cluster löschen.

Löschen aller Ihrer Cluster

So löschen Sie einen Cluster in der Amazon DocumentDB-Konsole:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, den Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Wenn Sie aufgefordert werden, einen endgültigen Snapshot zu erstellen?, wählen Sie Ja oder Nein aus.
6. Wenn Sie im vorherigen Schritt Yes (Ja) gewählt haben, geben Sie unter Final snapshot name (Endgültiger Snapshot-Name) den Namen Ihres endgültigen DB-Snapshots ein.
7. Wählen Sie Delete (Löschen).

> **Note**
>
> Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um die serviceverknüpfte Rolle `AWSServiceRoleForDocDB-Elastic` zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im AWS Benutzerhandbuch für Identity and Access Management.

Amazon-Documententententent

Die Überwachung IhrerAWS Dienste ist ein wichtiger Bestandteil, damit Ihre Systeme gesund bleiben und optimal funktionieren. Es empfiehlt sich, Überwachungsdaten aller Bestandteile IhrerAWS - Lösung zu sammeln, damit Sie auftretende Fehler oder Beeinträchtigungen leichter beheben können. Bevor Sie mit der Überwachung IhrerAWS Lösungen beginnen, empfehlen wir Ihnen, die folgenden Fragen zu berücksichtigen und Antworten zu formulieren:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden Sie überwachen?
- Wie oft werden Sie diese Ressourcen überwachen?
- Welche Überwachungstools werden verwendet?
- Wer ist für die Überwachung verantwortlich?
- Wer ist zu benachrichtigen und auf welche Weise, wenn etwas schief geht?

Um Ihre aktuellen Leistungsmuster zu verstehen, Leistungsanomalien zu identifizieren und Methoden zur Behebung von Problemen zu formulieren, sollten Sie grundlegende Leistungskennzahlen für verschiedene Zeiten und unter unterschiedlichen Lastbedingungen festlegen. Während Sie IhreAWS Lösung überwachen, empfehlen wir Ihnen, Ihre historischen Überwachungsdaten zu speichern, damit Sie später darauf future und Ihre Ausgangswerte festlegen können.

Die zulässigen Werte für Leistungsmetriken sind im Allgemeinen davon abhängig, wie die Ausgangsbasis aussieht und wofür die Anwendung gedacht ist. Prüfen Sie, ob dauerhafte oder tendenzielle Abweichungen von Ihrer Ausgangsbasis vorliegen. Im Folgenden werden Hinweise zu bestimmten Arten von Kennzahlen gegeben:

- Hohe CPU- oder RAM-Nutzung — Hohe Werte für die CPU- oder RAM-Nutzung können angemessen sein, wenn sie der Zielsetzung Ihrer Anwendung entsprechen (z. B. in Bezug auf Durchsatz oder Gleichzeitigkeit) und erwartet werden.
- Nutzung des Datenträgerplatzes — Überprüfen Sie die Nutzung des Datenträgerplatzes (`VolumeBytesUsed`), wenn konsistent 85 Prozent oder mehr des gesamten Datenträgerplatzes belegt werden. Legen Sie fest, ob Sie Daten aus dem Speicher-Volume löschen oder Daten in ein anderes System archivieren können, um Platz zu schaffen. Weitere Informationen erhalten Sie unter Amazon DocumentDB-Speicher und Kontingente und Limits für Amazon DocumentDB .

- Netzwerkdatenverkehr — Wenden Sie sich an Ihren Systemadministrator, um zu erfahren, welcher Durchsatz für Ihr Domänennetzwerk und Ihre Internetverbindung erwartet wird. Überprüfen Sie den Netzwerkdatenverkehr, wenn der Durchsatz dauerhaft unter dem erwarteten Wert liegt.
- Datenbankverbindungen — Ziehen Sie eine Einschränkung der Datenbankverbindungen in Betracht, wenn bei einer großen Anzahl von Benutzerverbindungen eine Abnahme der Instance-Leistung und der Reaktionszeit zu erkennen ist. Die optimale Anzahl der Benutzerverbindungen für Ihre Instance ist von der Instance-Klasse und der Komplexität der Operationen abhängig, die ausgeführt werden.
- IOPS-Metriken — Die erwarteten Werte für IOPS-Metriken sind von der Datenträgerspezifikation und der Serverkonfiguration abhängig. Verwenden Sie die Basiswerte als typische Werte. Prüfen Sie, ob dauerhafte Abweichungen von den Werten Ihrer Ausgangsbasis vorliegen. Für eine optimale IOPS-Leistung stellen Sie sicher, dass Ihr typisches Working Set in den Speicher passt, um Lese- und Schreibvorgänge zu minimieren.

Amazon DocumentDB CloudWatch Sie können Amazon DocumentDB DocumentDB-Metriken mithilfe verschiedener Tools einsehen, darunter die Amazon DocumentDB DocumentDB-KonsoleAWS CLI,CloudWatch API und Performance Insights.

Themen

Überwachung des Status eines Amazon DocumentDB-Clusters

Der Status eines Clusters zeigt den Zustand des Clusters an. Sie können den Status eines Clusters über die Amazon DocumentDBAWS CLI`describe-db-clusters`

Themen

Cluster-Statuswerte

In der folgenden Tabelle sind die gültigen Werte für den Cluster-Status aufgeführt.

Cluster-Status	Beschreibung
`active`	Der Cluster ist aktiv. Dieser Status gilt nur für elastische Cluster.
`available`	Der Cluster ist stabil und verfügbar. Dieser Status gilt nur für instancebasierte Cluster.
`backing-up`	Der Cluster wird derzeit gesichert.
`creating`	Der Cluster wird gerade erstellt. Ist während der Erstellung unzugänglich.
`deleting`	Der Cluster wird gerade gelöscht. Ist während des Löschvorgangs nicht zugreifbar.
`failing-over`	Es wird gerade ein Failover von der primären Instance in ein Amazon DocumentDB
`inaccessible-encryption-credentials`	Es ist kein Zugriff auf den AWS KMS-Schlüssel möglich, der zum Ver- oder Entschlüsseln des Clusters verwendet wird.

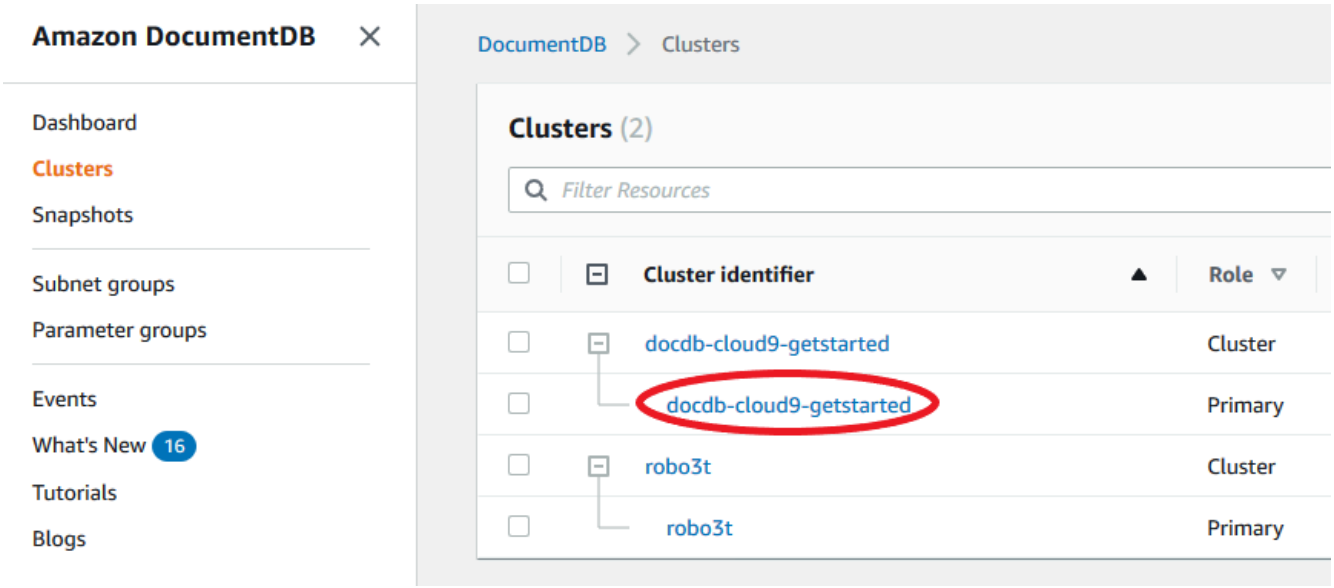
Cluster-Status	Beschreibung
`maintenance`	Ein Wartungsupdate wird auf den Cluster angewende t. Dieser Status wird bei Wartungen auf der Ebene des Clusters verwendet, die von Amazon DocumentDB
`migrating`	Es wird gerade ein Cluster-Snapshot in einen Cluster wiederhergestellt.
`migration-failed`	Eine Migration ist fehlgesch lagen.
`modifying`	Der Cluster wird aufgrund einer Kundenanfrage, den Cluster zu ändern, geändert.
`renaming`	Der Cluster wird aufgrund einer Kundenanfrage auf Instance-Umbenennung umbenannt.
`resetting-master-credentials`	Die Master-Anmeldeinfo rmationen für den Cluster werden auf eine Kundenanf rage auf Zurücksetzung hin zurückgesetzt.
`upgrading`	Die Cluster-Engine wird auf eine neue Version aktualisiert.

Den Status eines Clusters überwachen

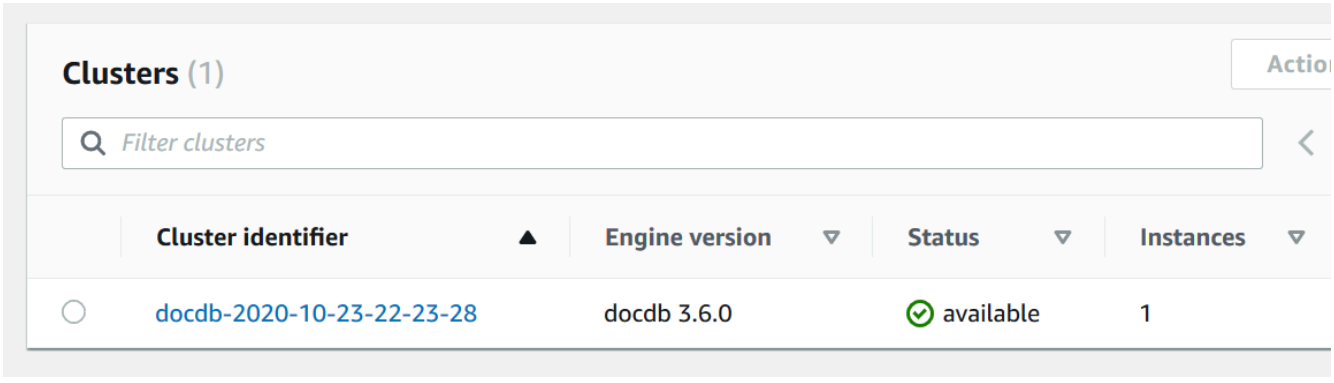
Using the AWS Management Console

Wenn Sie die AWS Management Console verwenden, um den Status eines Clusters zu bestimmen, gehen Sie folgendermaßen vor.

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon DocumentDB https://console.aws.amazon.com/docdb
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster).
3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances sind unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Suchen Sie in der Spalte Cluster-ID den Namen der Instanz, an der Sie interessiert sind. Um dann den Status der Instance zu ermitteln, lesen Sie in dieser Zeile die Spalte Status durch, wie unten dargestellt.



Using the AWS CLI

Wenn Sie die AWS CLI verwenden, um den Status eines Clusters zu bestimmen, nutzen Sie die Operation `describe-db-clusters`. Der folgende Code ermittelt den Status des Clusters `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Überwachung des Status einer Amazon DocumentDB DocumentDB-Instance

Amazon DocumentDB bietet Informationen über den aktuellen Zustand jeder konfigurierten Instance in der Datenbank.

Es gibt drei Arten von Status, die Sie für eine Amazon DocumentDB DocumentDB-Instance einsehen können:

- Instanzstatus: Dieser Status wird in der Spalte Status der Cluster-Tabelle in der angezeigtAWS Management Console und zeigt den aktuellen Lebenszykluszustand der Instance. Die in der Spalte

Status angezeigten Werte werden aus dem`Status` Feld der`DescribeDBCluster` API-Antwort abgeleitet.

- Integritätsstatus der Instanz: Dieser Status wird in der Spalte Instanzintegrität der Clustertabelle in derAWS Management Console und zeigt an, ob die Datenbank-Engine, die für die Verwaltung und den Abruf von Daten zuständige Komponente, läuft. Die in der Spalte Instanzstatus angezeigten Werte basieren auf derCloudWatch`EngineUptime` Amazon-Systemmetrik.
- Wartungsstatus: Dieser Status wird in der Spalte Wartung der Clustertabelle in der angezeigtAWS Management Console und gibt den Status aller Wartungsereignisse an, die auf eine Instance angewendet werden müssen. Der Wartungsstatus ist unabhängig vom Status der anderen Instanz und wird von der`PendingMaintenanceAction` API abgeleitet. Für weitere Informationen zum Wartungszeitfenstern für Amazon DocumentDB

Themen

Instance-Statuswerte

Die folgende Tabelle führt die möglichen Statuswerte für Instances auf und gibt an, wie welcher Status berechnet wird. Die Tabelle gibt an, ob Sie für Instance und Speicher, nur für Speicher oder weder für Instance noch Speicher zahlen müssen. Für alle Instance-Status wird immer die Sicherungsnutzung berechnet.

Instance-Status	Berechnet	Beschreibung
`available`	Berechnet	Die Instance ist stabil und verfügbar.
`backing-up`	Berechnet	Die Instance wird derzeit gesichert.
`configuring-log-exports`	Berechnet	Das Veröffentlichen von Protokolldateien an AmazonCloudWatch Logs wird aktiviert oder deaktiviert.

Instance-Status	Berechnet	Beschreibung
`creating`	Nicht berechnet	Die Instance wird gerade erstellt. Während die Instance erstellt wird, kann nicht auf sie zugegriffen werden.
`deleting`	Nicht berechnet	Die Instance wird gerade gelöscht.
`failed`	Nicht berechnet	Die Instance befindet sich im Fehlerzustand und Amazon DocumentDB Führen Sie zur Wiederherstellung der Daten einepoint-in-time Wiederherstellung auf den neuesten wiederherstellbaren Zeitpunkt der Instance durch.
`inaccessible-encryption-credentials`	Nicht berechnet	Der Schlüssel AWS KMS, mit dem die Instance verschlüsselt oder entschlüsselt wird, konnte nicht aufgerufen werden.
`incompatible-network`	Nicht berechnet	Amazon DocumentDB Dieser Status kann beispielsweise eintreten, wenn alle IP-Adressen in einem Subnetz belegt sind und Amazon DocumentDB
`maintenance`	Berechnet	Amazon DocumentDB Dieser Status wird bei Wartungen auf der Ebene der Instance verwendet, die von Amazon DocumentDB Wir evaluieren derzeit Möglichkeiten, Kunden über diesen Status auch weitere Wartungsaktionen kenntlich zu machen.
`modifying`	Berechnet	Die Instance wird aufgrund einer Anforderung zur Änderung der Instance geändert.
`rebooting`	Berechnet	Die Instance wird aufgrund einer Anfrage oder eines Amazon DocumentDB

Instance-Status	Berechnet	Beschreibung
`renaming`	Berechnet	Die Instance wird aufgrund einer Anforderung zur Umbenennung umbenannt.
`resetting-master-credentials`	Berechnet	Die Master-Anmeldeinformationen für die Instance werden aufgrund einer Anforderung zum Zurücksetzen zurückgesetzt.
`restore-error`	Berechnet	Bei dem Versuch, die Instance anhand einespoint-in-time oder eines Snapshots wiederherzustellen, ist ein Fehler aufgetreten.
`starting`	Berechnet für Speicher	Die Instance wird gestartet.
`stopped`	Berechnet für Speicher	Die Instance wird angehalten.
`stopping`	Berechnet für Speicher	Die Instance wird gestoppt.
`storage-full`	Berechnet	Die Instance hat seine Speicherkapazitätszuteilung erreicht. Dieser Status ist kritisch und sollte sofort behoben werden. Sie müssen den verfügbaren Speicherplatz hochskalieren, indem Sie die Instance entsprechend ändern. Richten SieCloudWatch Amazon-Alarme ein, damit Sie gewarnt werden, wenn der Speicherplatz knapp wird.

Überwachung des Instanzstatus mithilfe desAWS Management Console oderAWS CLI

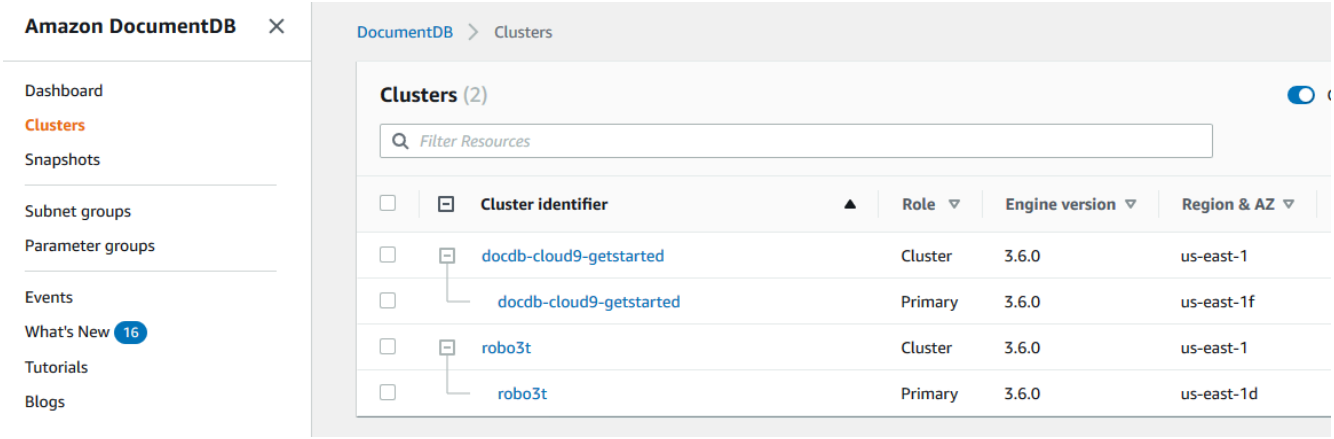
Verwenden Sie dasAWS Management Console oderAWS CLI, um den Status Ihrer Instance zu überwachen.

Using the AWS Management Console

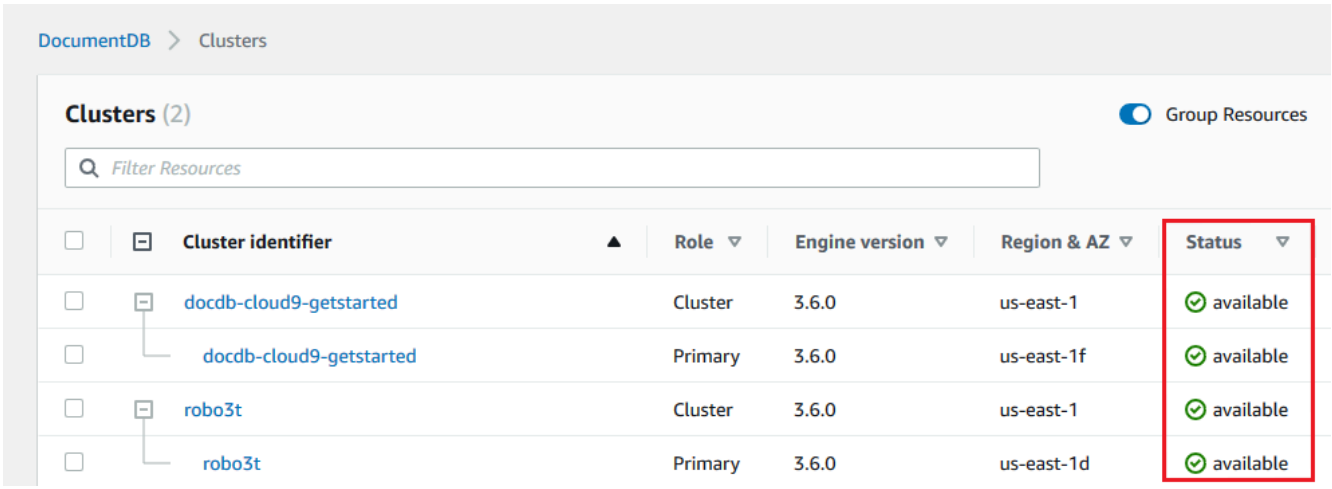
Wenn Sie die AWS Management Console verwenden, um den Status eines Clusters zu bestimmen, gehen Sie folgendermaßen vor.

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon DocumentDB https://console.aws.amazon.com/docdb
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster).

> **Note**
>
> Beachten Sie, dass im Cluster-Navigationsfeld in der Spalte Cluster-ID sowohl Cluster als auch Instances angezeigt werden. Instanzen werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



3. Suchen Sie den Namen der Instanz, an der Sie interessiert sind. Um den Status der Instance zu ermitteln, prüfen Sie die Zeile in die Spalte Status.



Using the AWS CLI

Wenn Sie die AWS CLI verwenden, um den Status eines Clusters zu bestimmen, nutzen Sie die Operation `describe-db-instances`. Der folgende Code ermittelt den Status der Instance `sample-cluster-instance-01`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
          --db-instance-identifier sample-cluster-instance-01  \
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Für Windows:

```
aws docdb describe-db-instances ^
          --db-instance-identifier sample-cluster-instance-01  ^
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
          [
              "sample-cluster-instance-01",
              "available"
          ]
      ]
```

Instance-Instance-Zustand

In der folgenden Tabelle sind die möglichen Integritätsstatuswerte für Instances aufgeführt. Die Spalte Instanzintegrität, die sich in der Cluster-Tabelle in der befindetAWS Management Console, zeigt, ob die Datenbank-Engine, die Komponente, die für das Speichern, Verwalten und Abrufen von Daten verantwortlich ist, normal funktioniert. Diese Spalte gibt auch an, ob die`EngineUptime` SystemmetrikCloudWatch, verfügbar in, den Integritätsstatus jeder Instanz anzeigt.

Instance-Zustand	Beschreibung
fehlerfrei	Die Datenbank-Engine läuft in der Amazon DocumentDB DocumentDB-Instance.
ungesund	Die Datenbank-Engine läuft nicht oder wurde vor weniger als einer Minute neu gestartet.

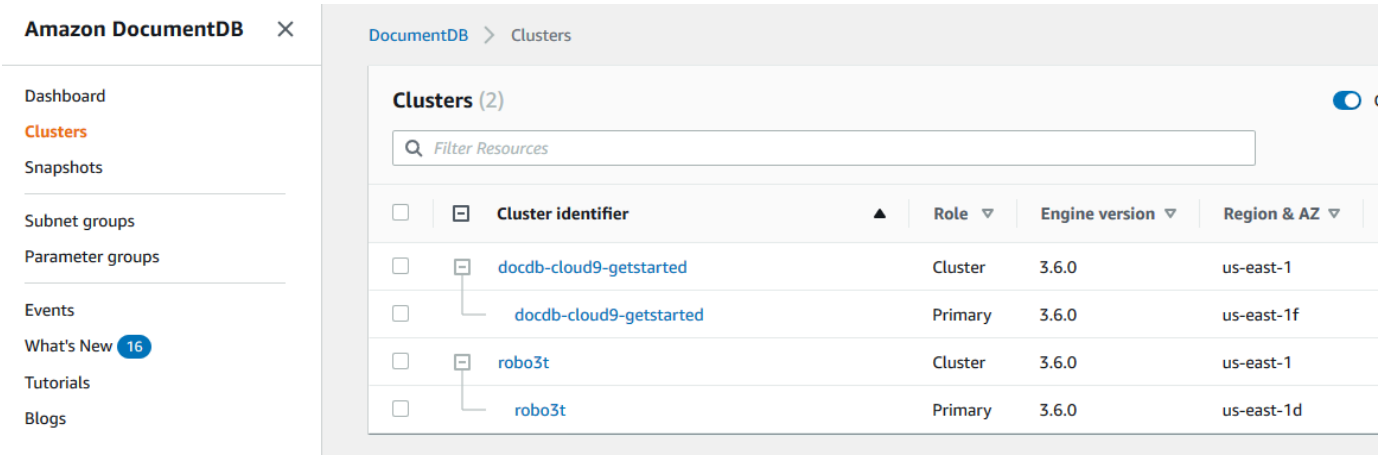
Überwachung des Zustands der Instanz mithilfe desAWS Management Console

Verwenden Sie dieAWS Management Console, um den Gesundheitsstatus Ihrer Instance zu überwachen.

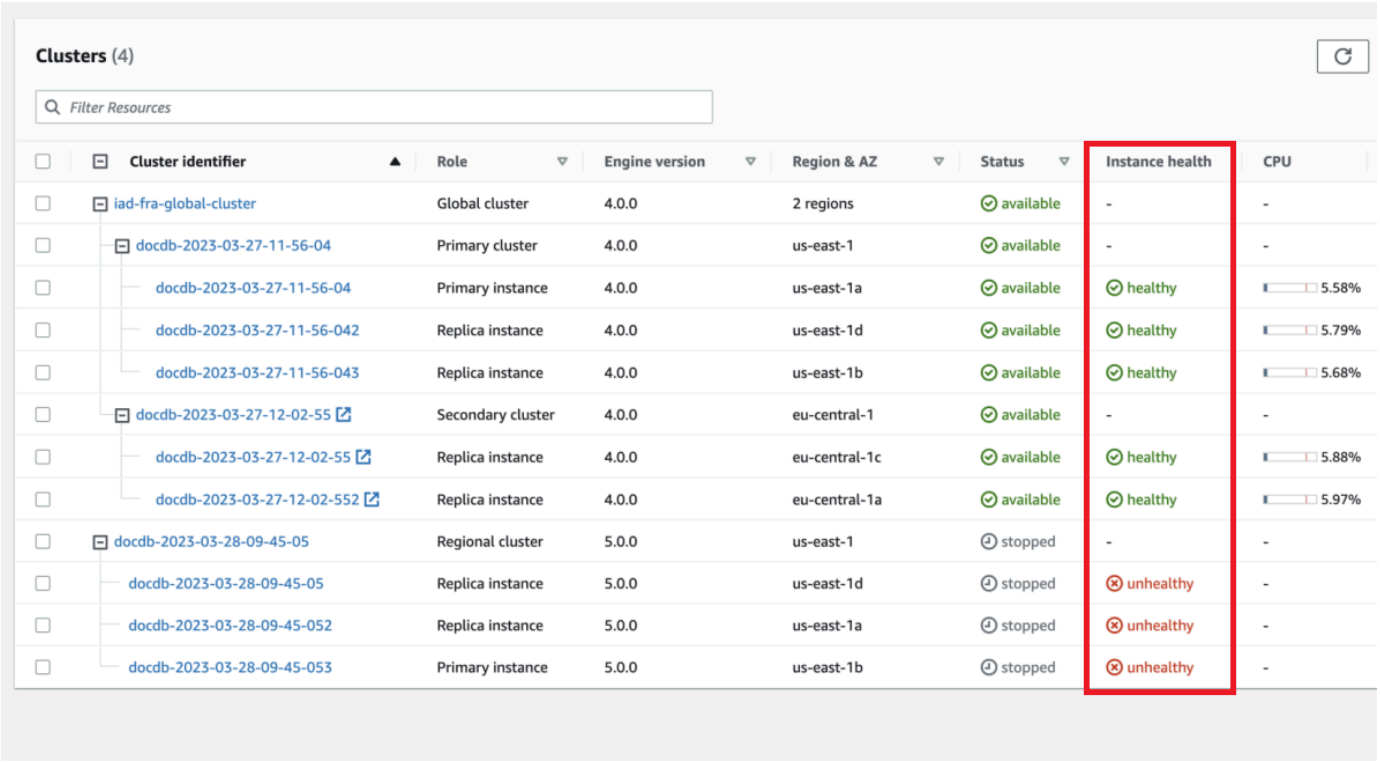
Gehen Sie bei der Verwendung der folgenden Schritte vorAWS Management Console, um den Integritätsstatus der Instance zu verstehen.

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon DocumentDB https://console.aws.amazon.com/docdb
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster).

> **Note**
>
> Im Navigationsfeld Cluster werden in der Spalte Cluster-Identifier sowohl Cluster als auch Instances angezeigt. Instanzen werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



3. Suchen Sie den Namen der Instanz, an der Sie interessiert sind. Um den Status der Instance zu ermitteln, lesen Sie dann in dieser Zeile die Spalte Instanzintegrität durch, wie in der folgenden Abbildung dargestellt:



> **Note**
>
> Die Integritätsstatusabfrage der Instanz erfolgt alle 60 Sekunden und basiert auf derCloudWatch`EngineUptime` Systemmetrik. Die Werte in der Spalte Instanzintegrität werden automatisch aktualisiert.

Amazon DocumentDB

Amazon DocumentDB Diese Empfehlungen bieten Anleitungen nach bewährten Methoden, indem sie Ihre Cluster- und Instance-Konfigurationen analysieren.

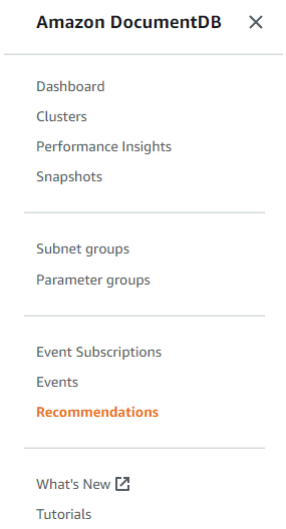
Ein Beispiel für diese Empfehlungen finden Sie in den folgenden Empfehlungen:

Typ	Beschreibung	Empfehlung	Zusätzliche Informationen
Eine -Instance	Cluster enthält nur eine Instanz	Leistung und Verfügbarkeit: Wir empfehlen, eine weitere Instance mit derselben Instance-Klasse in einer anderen Availability Zone hinzuzufügen.	Amazon DocumentDB Hochverfügbarkeit und Replikation

Amazon DocumentDB Amazon DocumentDB

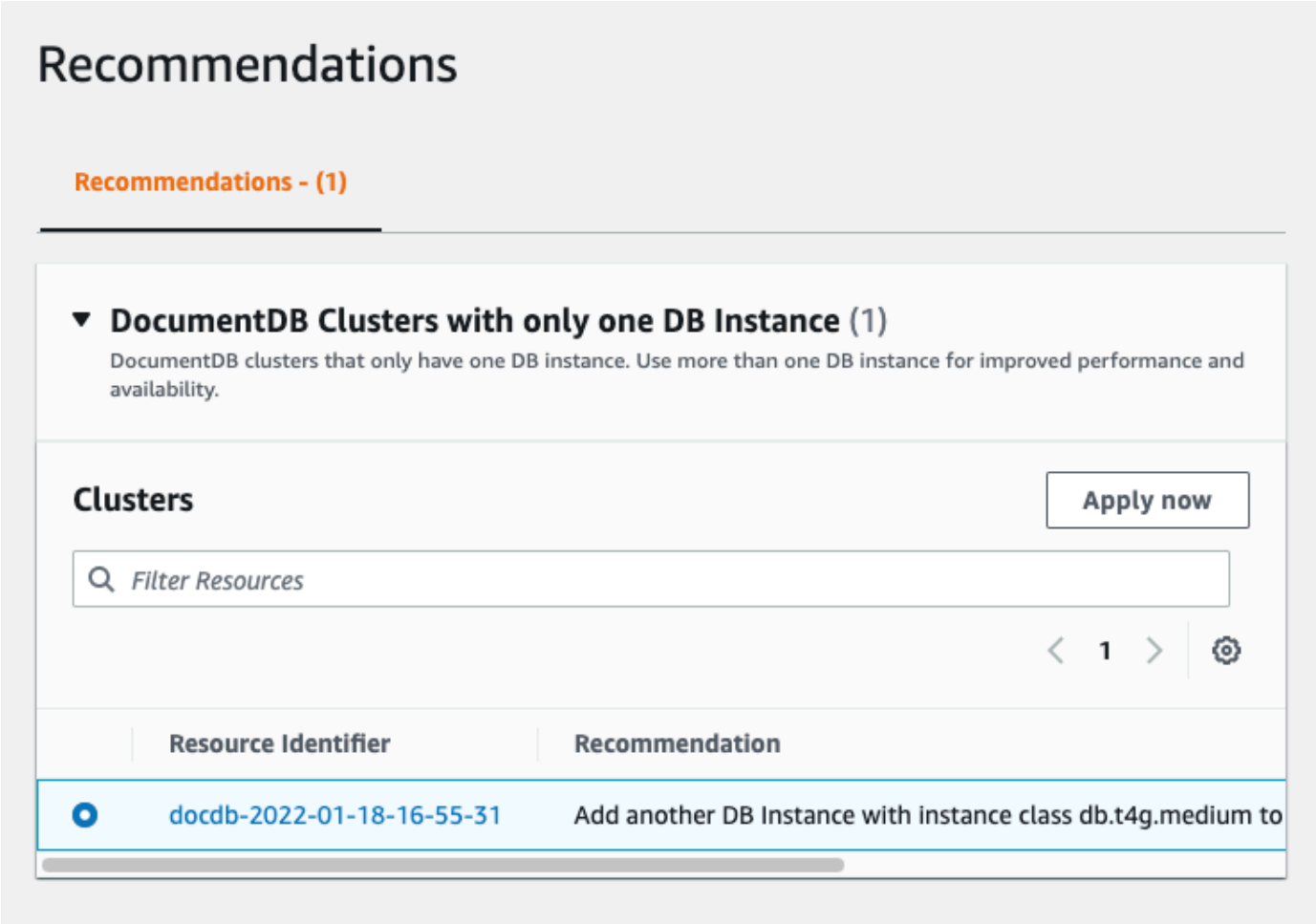
Um die Empfehlungen von Amazon DocumentDB einzusehen und entsprechende Maßnahmen zu ergreifen

1. Melden Sie sich bei derAWS Management Console an und öffnen Sie die Amazon DocumentDB https://console.aws.amazon.com/docdb
2. Wählen Sie im Navigationsbereich Empfehlungen aus:



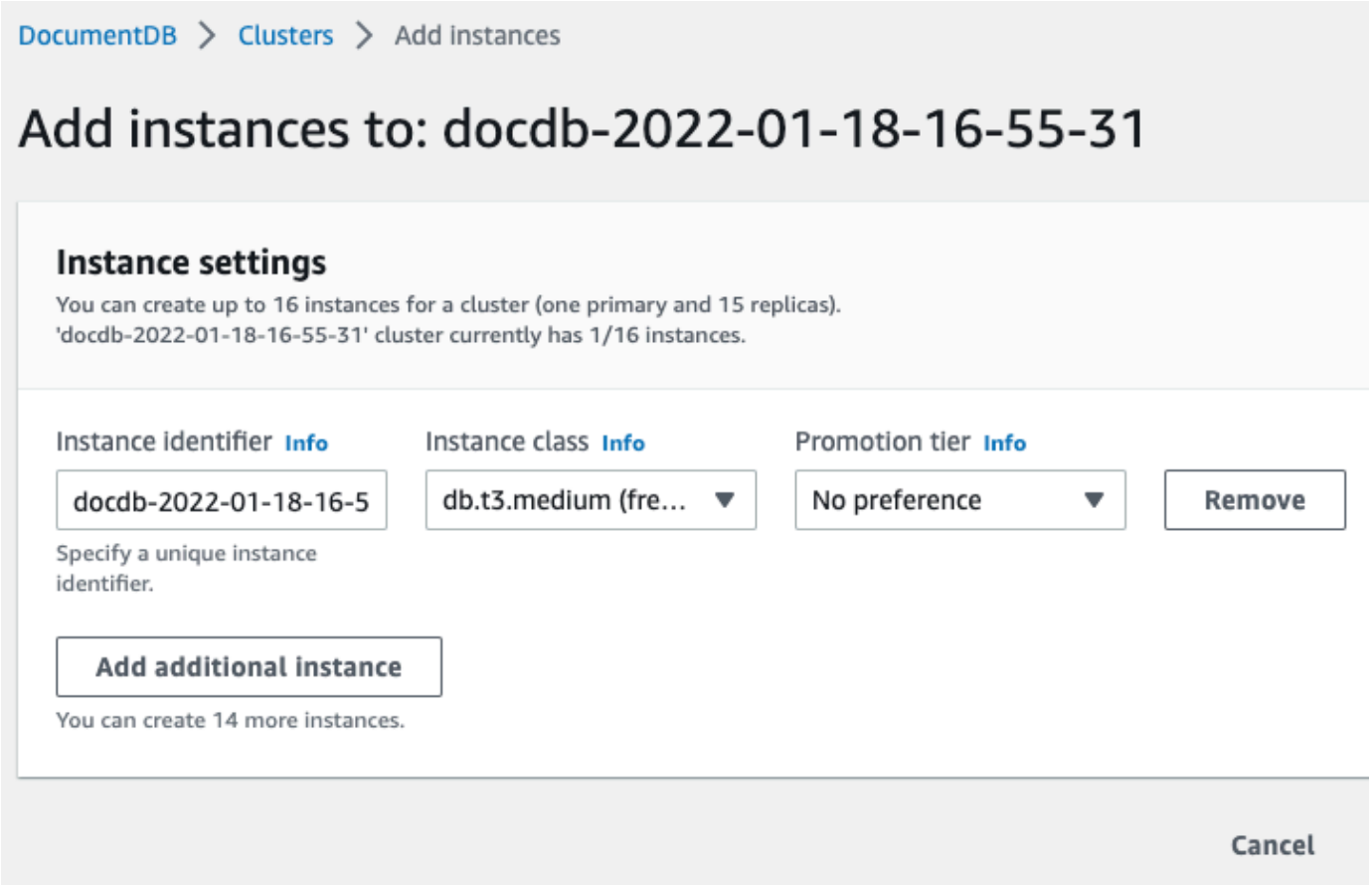
3. Erweitern Sie im Dialogfeld „Empfehlungen“ den gewünschten Bereich und wählen Sie die empfohlene Aufgabe aus.

   Im folgenden Beispiel gilt die empfohlene Aufgabe für einen Amazon DocumentDB-Cluster mit nur einer Instance. Es wird empfohlen, eine weitere Instanz hinzuzufügen, um die Leistung und Verfügbarkeit zu verbessern.



4. Klicken Sie auf Jetzt bewerben.

   In diesem Beispiel wird das Dialogfeld Instanzen hinzufügen angezeigt:



5. Ändern Sie die Einstellungen Ihrer neuen Instanz und klicken Sie auf Erstellen.

Verwenden von Amazon DocumentDB DocumentDB-Event-Abonnements

Amazon DocumentDB verwendet Amazon Simple Notification Service (Amazon SNS), um Benachrichtigungen zu senden, wenn ein Amazon DocumentDB DocumentDB-Ereignis stattfindet. Diese Benachrichtigungen können jedes von Amazon SNS für einen unterstützte Format aufweisenAWS-Region, wie zum Beispiel eine E-Mail, eine SMS, eine E-Mail, eine SMS oder einen Anruf an einen HTTP-Endpunkt.

Amazon DocumentDB gruppiert diese Ereignisse in Kategorien, die Sie abonnieren können, um Benachrichtigungen zu erhalten, wenn ein Ereignis in dieser Kategorie stattfindet. Sie können eine Ereigniskategorie für eine Instance, ein DB-Cluster-Snapshot, einen DB-Cluster-Snapshot, einen DB-Cluster-Snapshot, einen DB-Cluster-Snapshot, einen DB Wenn Sie zum Beispiel die Kategorie „Backup“ für eine bestimmte Instance abonnieren, werden Sie immer dann benachrichtigt, wenn ein

Backup-bezogenes Ereignis eintritt, das sich auf Ihre Instance auswirkt. Außerdem erhalten Sie eine Benachrichtigung, wenn ein Ereignisabonnements wird.

Ereignisse treten auf Cluster- und auf Instance-Ebene auf. Daher erhalten Sie keine Ereignisse, wenn Sie einen Cluster oder eine Instance abonnieren.

Ereignisabonnements werden an die Adressen gesendet, die Sie beim Erstellen des Abonnements angeben. Sie können mehrere verschiedene Abonnements erstellen, beispielsweise ein Abonnement, das alle Ereignisbenachrichtigungen empfängt, und ein anderes Abonnement, das nur kritische Ereignisse für Ihre Produktions-Instances enthält. Sie können die Benachrichtigung ganz einfach deaktivieren, ohne ein Abonnement zu löschen. Setzen Sie dazu das Optionsfeld Aktiviert in der Amazon DocumentDB DocumentDB-Konsole auf Nein.

> **Important**
>
> Amazon DocumentDB garantiert nicht die Reihenfolge der Ereignisse, die in einem Ereignisstrom gesendet werden. Die Reihenfolge der Ereignisse kann sich ändern.

Amazon DocumentDB verwendet den Amazon-Ressourcennamen (ARN) eines Amazon SNS SNS-Themas, um die einzelnen Abonnements zu ermitteln. Die Amazon DocumentDB DocumentDB-Konsole erstellt einen ARN für Sie, wenn Sie ein Abonnement erstellen.

Die Fakturierung für Amazon-Ereignisabonnements erfolgt über Amazon SNS. Bei Verwendung von Ereignisbenachrichtigungen fallen Amazon-SNS-Gebühren an. Weitere Informationen finden Sie unter Amazon Simple Notification — Preise. Abgesehen von den Amazon SNS SNS-Gebühren berechnet Amazon DocumentDB keine Event-Abonnements.

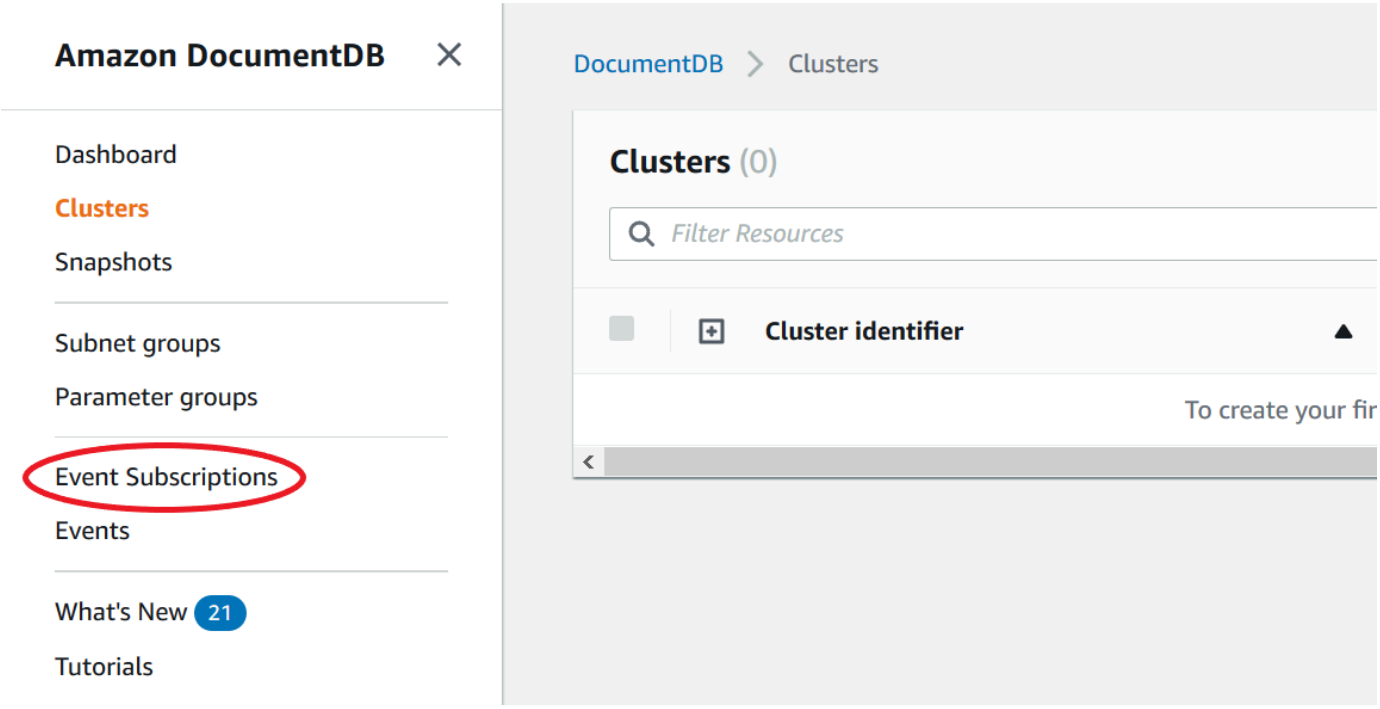
Themen

- Abonnieren von Amazon DocumentDB DocumentDB-Ereignisabonnements
- Verwaltung von Amazon DocumentDB DocumentDB-Abonnements für Eventbenachrichtigungen
- Amazon DocumentDB DocumentDB-Ereigniskategorien und Nachrichten

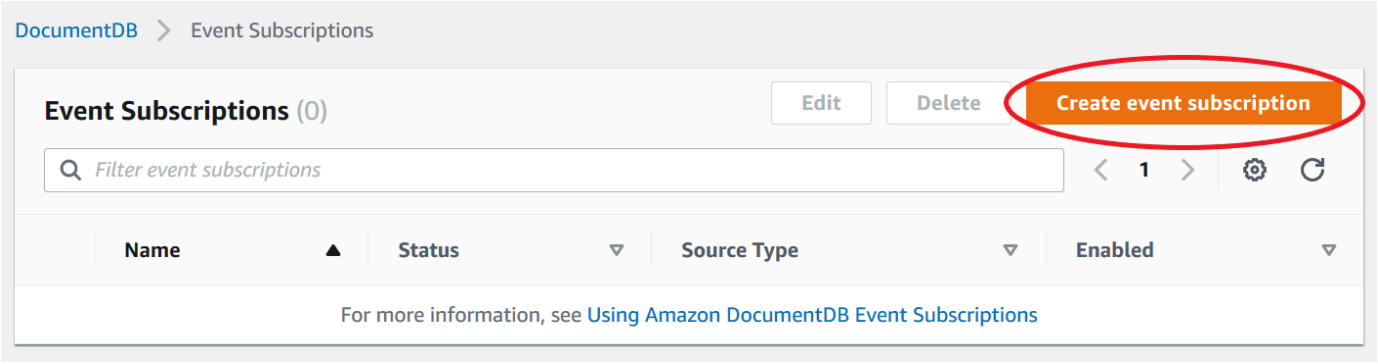
Abonnieren von Amazon DocumentDB DocumentDB-Ereignisabonnements

Sie können die Amazon DocumentDB DocumentDB-Konsole wie folgt verwenden, um Veranstaltungsabonnements zu abonnieren:

1. Melden Sie sich unter https://console.aws.amazon.com/docdb bei der AWS Management Console an.
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus.

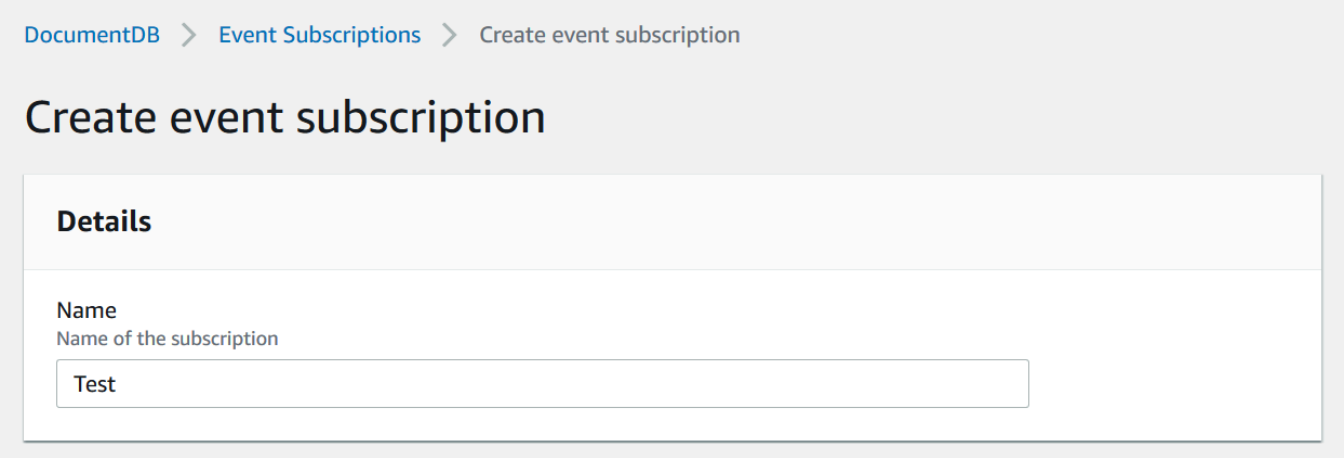


3. Wählen Sie im Bereich Ereignisabonnements Ereignisabonnement erstellen aus.

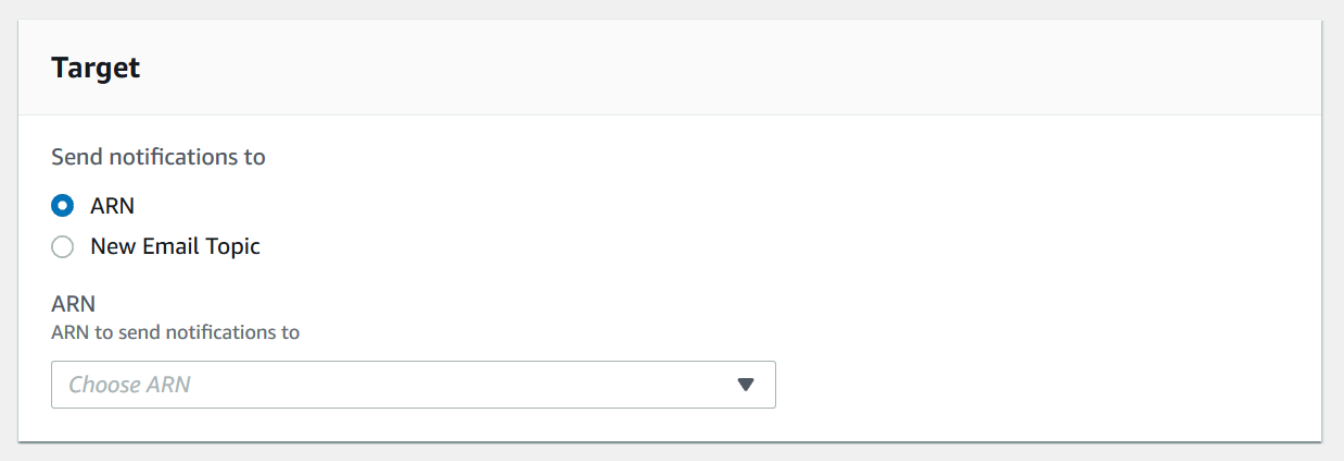


4. Gehen Sie im Dialogfeld Ereignisabonnement erstellen wie folgt vor:

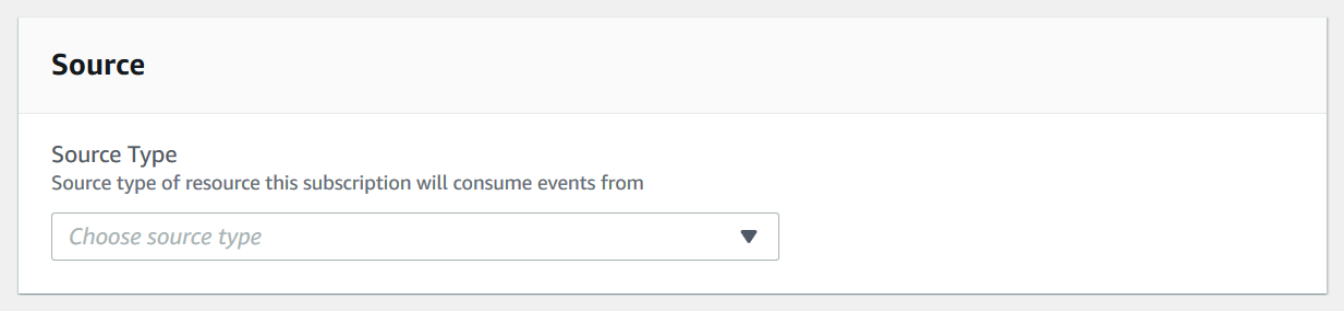
   - Geben Sie unter Name einen Namen für das Abonnement für Ereignisbenachrichtigungen ein.



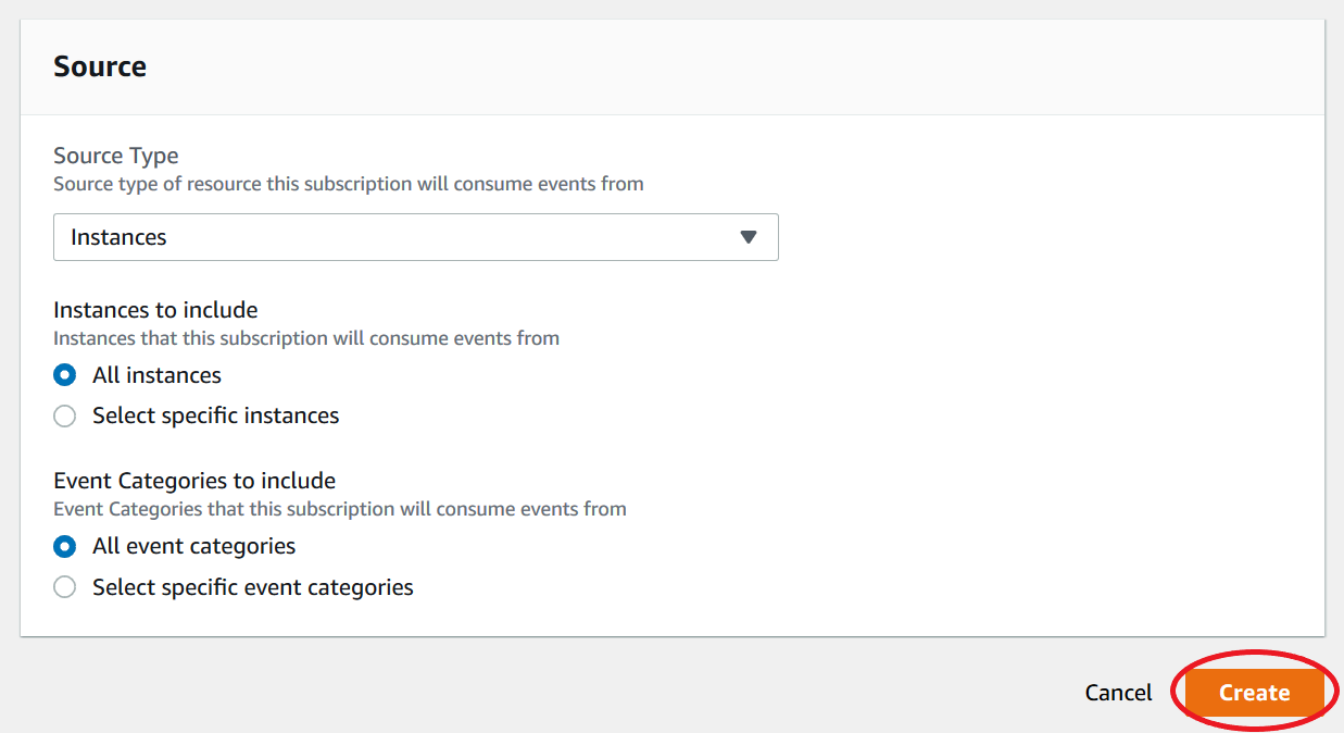
- Wählen Sie für Target aus, an wen Sie Benachrichtigungen senden möchten. Sie können einen vorhandenen ARN auswählen oder Neues E-Mail-Thema wählen, um den Namen eines Themas und eine Liste von Empfängern einzugeben.



- Wählen Sie unter Quelle einen Quelltyp aus. Je nach ausgewähltem Quelltyp wählen Sie die Ereigniskategorien und die Quellen aus, von denen Sie Ereignisbenachrichtigungen erhalten möchten.



- Wählen Sie Create (Erstellen) aus.

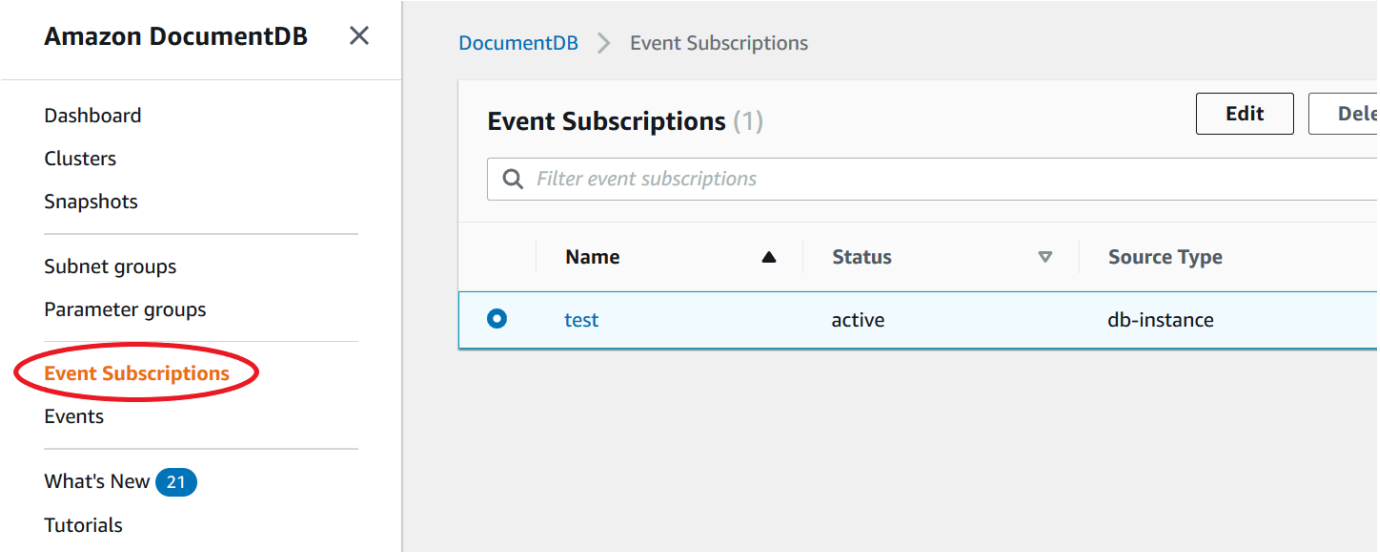


Verwaltung von Amazon DocumentDB DocumentDB-Abonnements für Eventbenachrichtigungen

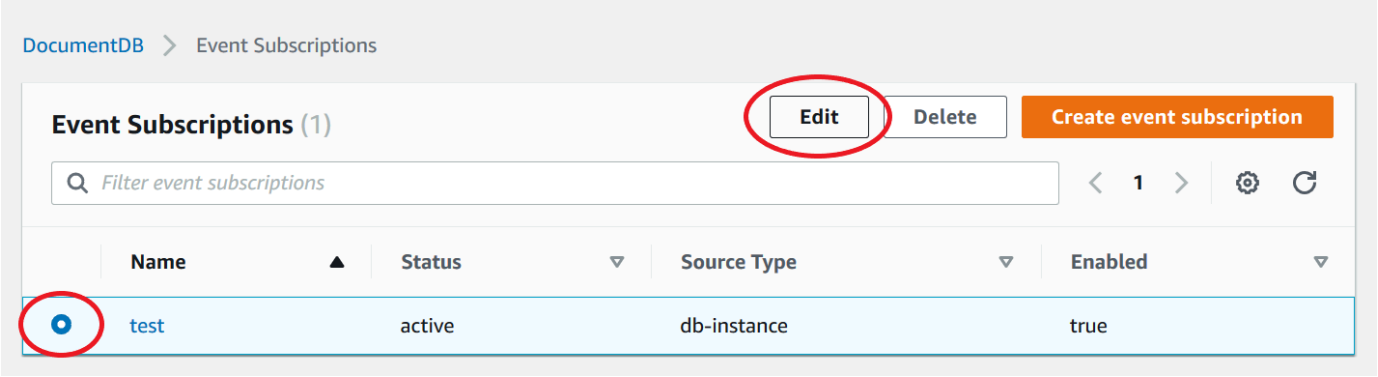
Wenn Sie im Navigationsbereich der Amazon DocumentDB DocumentDB-Konsole die Option Event-Abonnements wählen, können Sie Abonnementkategorien und eine Liste Ihrer aktuellen Abonnements einsehen. Sie können auch ein bestimmtes Abonnement ändern oder löschen.

So modifizieren Sie Ihre aktuellen Abonnements für Amazon-DocumentDB-Ereignisbenachrichtigungen

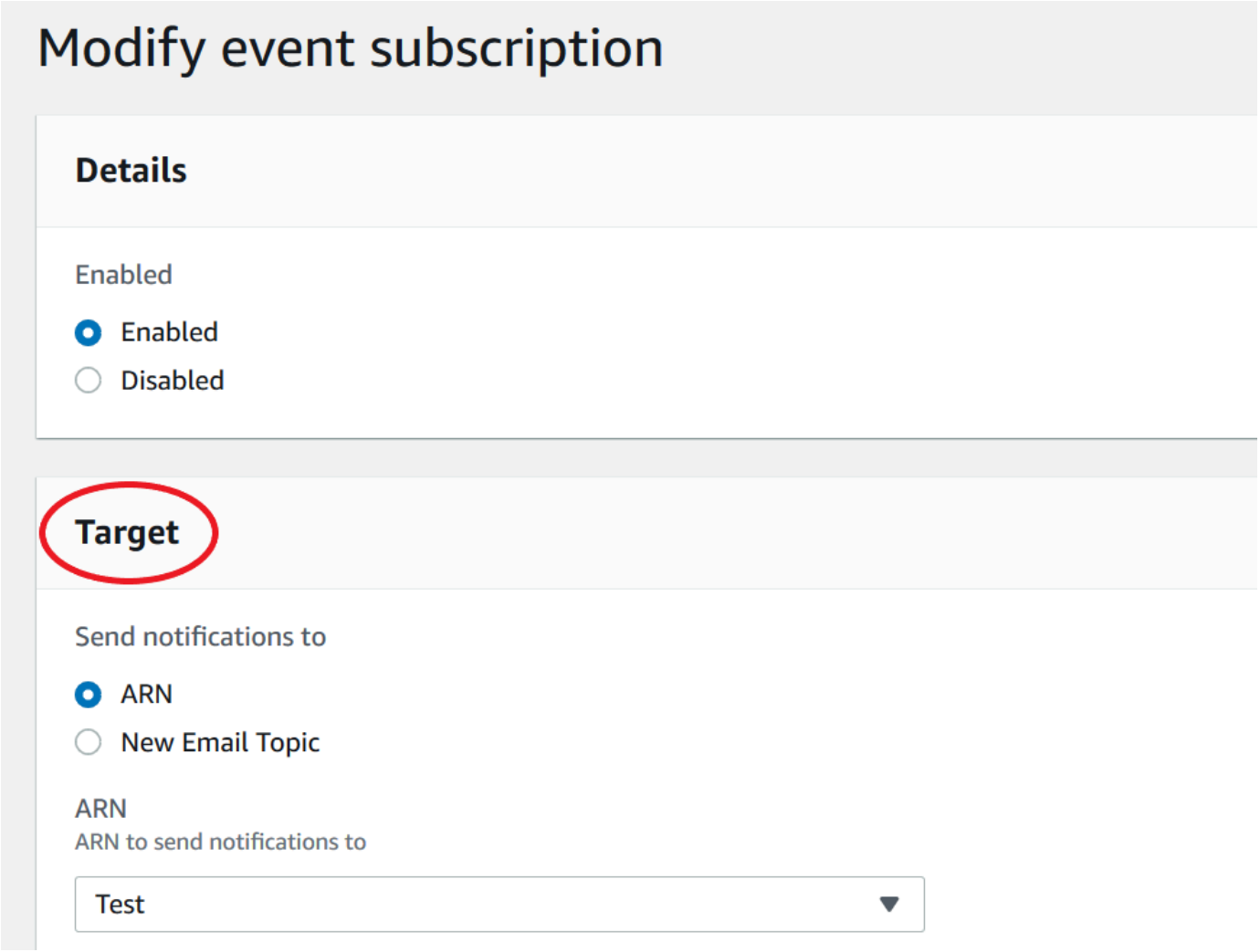
1. Melden Sie sich unter https://console.aws.amazon.com/docdb bei der AWS Management Console an.
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus. Im Bereich Ereignisabonnements werden all Ihre Abonnements für Ereignisbenachrichtigungen angezeigt.



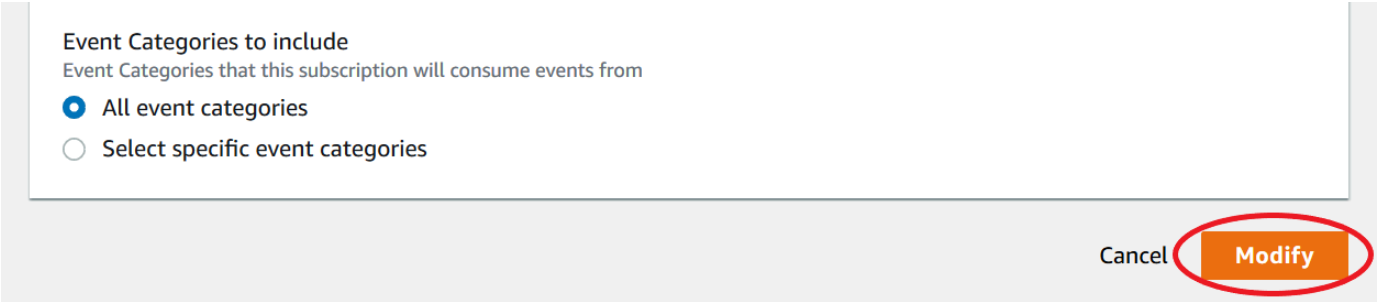
3. Wählen Sie im Bereich Ereignisabonnements das Abonnement, das Sie modifizieren möchten, und klicken Sie auf Bearbeiten.



4. Nehmen Sie Ihre Änderungen am Abonnement im Bereich Ziel oder Quelle vor. Sie können Quell-IDs hinzufügen oder entfernen, indem Sie diese im Abschnitt Quelle aktivieren oder deaktivieren.



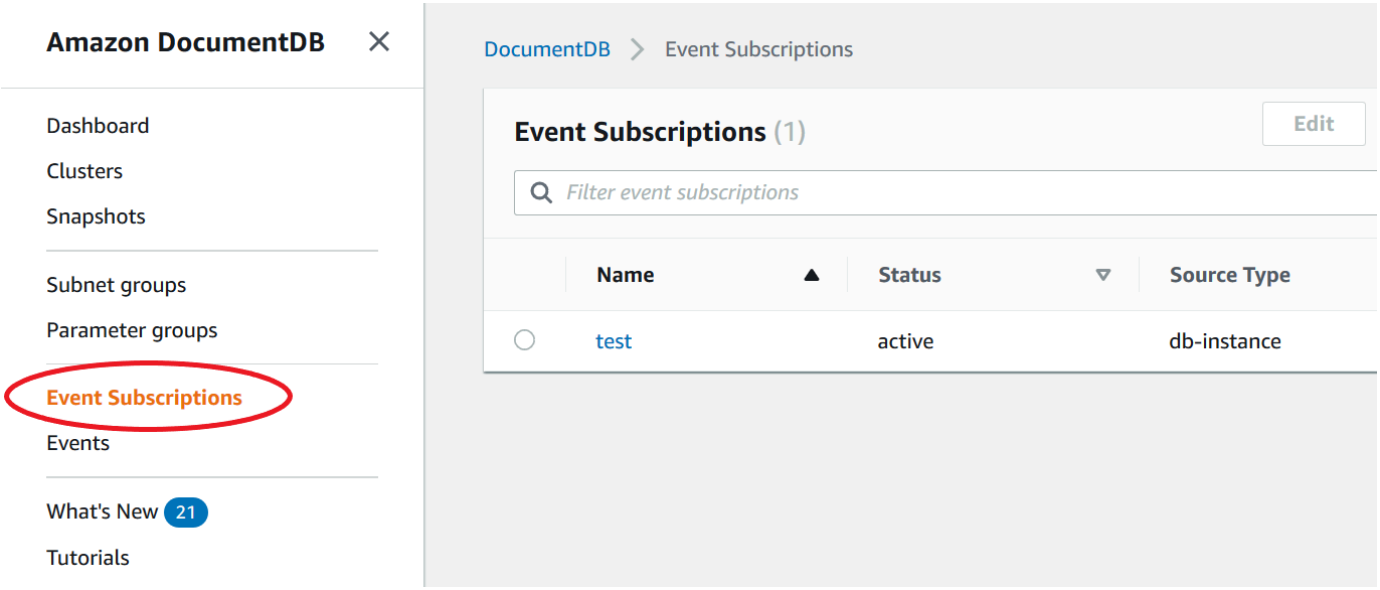
5. Wählen Sie Ändern aus. In der Amazon DocumentDB DocumentDB-Konsole wird die Änderung des Abonnements angezeigt.



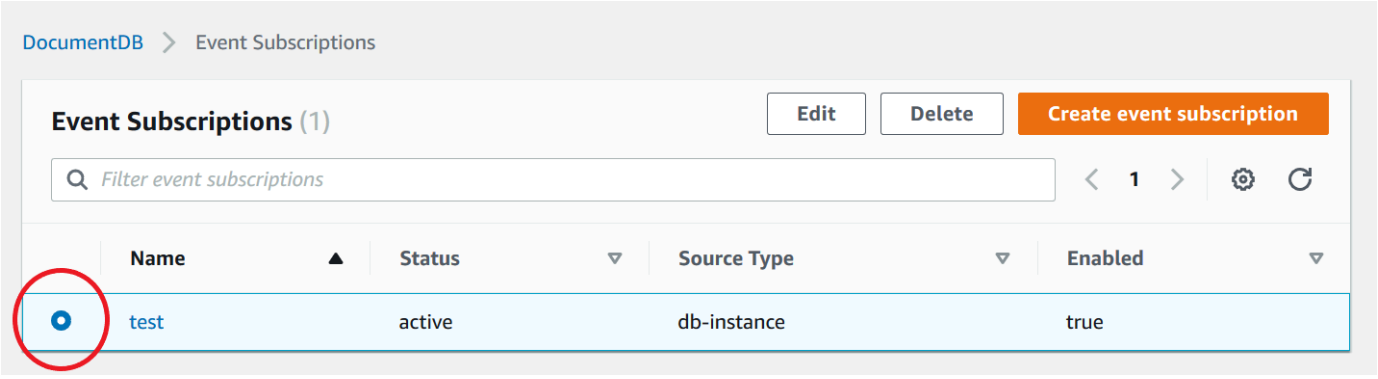
Löschen eines Abonnements für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen

Sie können ein Abonnement löschen, wenn Sie es nicht mehr benötigen. Alle Abonnenten des Themas erhalten dann keine weiteren Ereignisbenachrichtigungen, die über dieses Abonnement ausgegeben wurden.

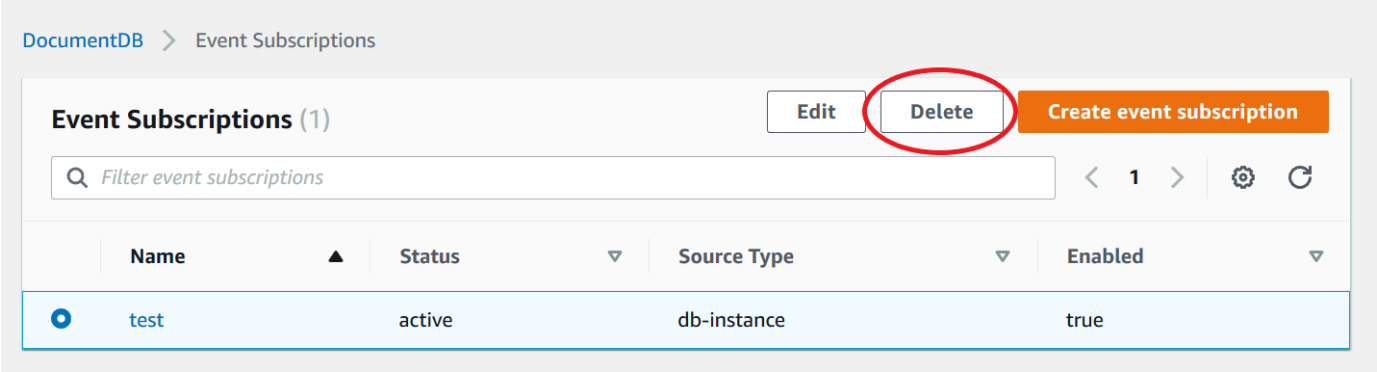
1. Melden Sie sich unter https://console.aws.amazon.com/docdb bei der AWS Management Console an.
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus.



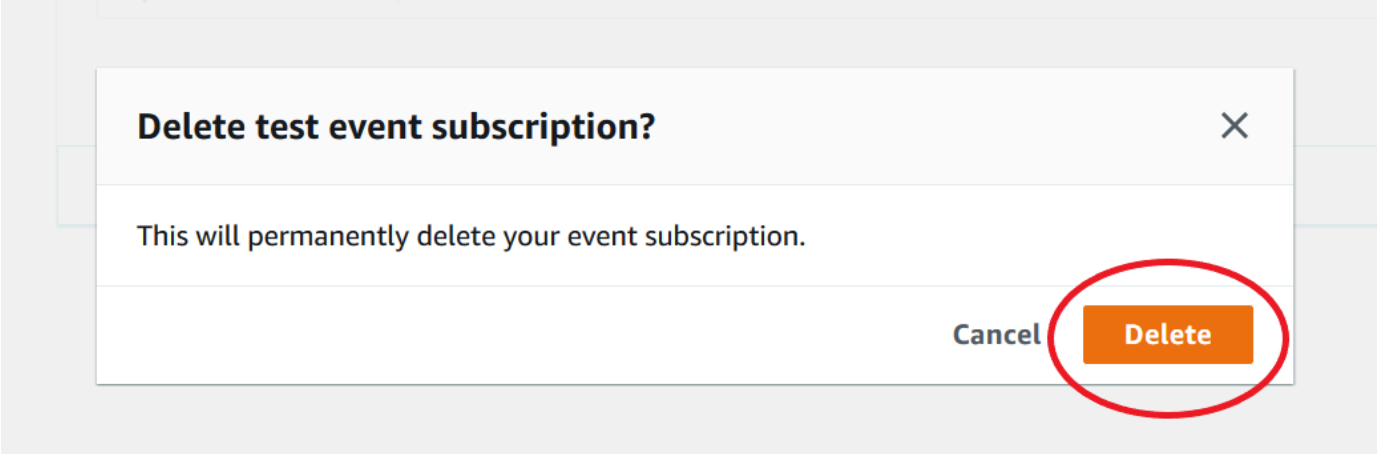
3. Wählen Sie im Bereich Ereignisabonnements das Abonnement, das Sie löschen möchten.



4. Wählen Sie Löschen.



5. In einem Popup-Fenster werden Sie gefragt, ob Sie diese Benachrichtigung dauerhaft löschen möchten. Wählen Sie Löschen.



Amazon DocumentDB DocumentDB-Ereigniskategorien und Nachrichten

Amazon DocumentDB generiert eine beträchtliche Anzahl von Ereignissen in Kategorien, die Sie mithilfe der -Konsole abonnieren können. Jede Kategorie gilt für einen Quelltyp, der eine Instance, ein Snapshot, eine Instance, ein Snapshot oder eine Instance sein kann.

> **Note**
>
> Amazon DocumentDB verwendet vorhandene Amazon RDS-Ereignisdefinitionen und -IDs.

Amazon DocumentDB DocumentDB-Ereignisse, die von Instances ausgehen

Kategorie	Beschreibung
Verfügbarkeit	Die Instance wurde neu gestartet.
Verfügbarkeit	Die Instance wird heruntergefahren.
Konfigurationsänderung	Die Änderung wird auf eine Instance-Klasse angewendet.
Konfigurationsänderung	Das Anwenden der Änderung auf eine Instanzklasse wurde abgeschlossen.
Konfigurationsänderung	Setzen Sie die Master-Zugangsdaten zurück.
Erstellung	Die Instance wurde erstellt.
Löschung	Instanz gelöscht
Ausfall	Die Instance ist aufgrund einer inkompatiblen Konfiguration oder eines zugrunde liegenden Speicherproblems ausgefallen. Beginne a point-in-time-restore für die Instanz.
Benachrichtigung	Die Instance wird angehalten.
Benachrichtigung	Die Instanz wurde gestartet.
Benachrichtigung	Die Instance wird gestartet, da sie die maximal zulässige Anhaltezeit überschritten hat.
Wiederherstellung	Wiederherstellung der Instance wurde gestartet. Die Wiederherstellungsdauer variiert je nach zu wiederherstellender Datenmenge.
Wiederherstellung	Wiederherstellung der Instance ist abgeschlossen.

Kategorie	Beschreibung
Ausführen von Sicherheits-Patches	Die Aktualisierung des Betriebssystems ist für Ihre Instance verfügbar. Informationen zum Anwenden von Aktualisierungen finden Sie unter Amazon DocumentDB.

Amazon DocumentDB DocumentDB-Ereignisse, die aus einem Cluster stammen

Kategorie	Beschreibung
Erstellung	Cluster erstellt
Löschung	Cluster gelöscht.
Failover	Erneut Werbung für die vorherige Primarver einigung.
Failover	Der Failover zur Instanz wurde abgeschlossen.
Failover	Failover zur DB-Instance gestartet: %s
Failover	Hat den gleichen AZ-Failover zur DB-Instance gestartet: %s
Failover	Cross-AZ-Failover zur DB-Instance gestartet: %s
Wartung	Der Cluster wurde gepatcht.
Wartung	Der Datenbank-Cluster befindet sich in einem Zustand, der nicht aktualisiert werden kann: %s
Benachrichtigung	Der Cluster wird angehalten.
Benachrichtigung	Der Cluster wird gestartet.
Benachrichtigung	Der Cluster-Snapshot ist fehlgeschlagen.

Kategorie	Beschreibung
Benachrichtigung	Der Cluster wird gestartet, da er die maximal zulässige Anhaltezeit überschritten hat.
Benachrichtigung	Cluster wurde von %s in %s umbenannt.

Amazon DocumentDB DocumentDB-Ereignisse, die aus einem Cluster-Snapshot stammen

In der folgenden Tabelle werden die Ereigniskategorie und die Ereignisse für den Quelltyp „Amazon DocumentDB-Cluster-Snapshot“ aufgeführt.

Kategorie	Beschreibung
backup	Erstellen eines manuellen Cluster-Snapshots.
backup	Manueller Cluster-Snapshot erstellt.
backup	Erstellen eines automatisierten Cluster-S napshots.
backup	Automatisierter Cluster-Snapshot erstellt.

Amazon DocumentDB DocumentDB-Ereignisse, die aus einer Parametergruppe stammen

Die folgende Tabelle zeigt den Ereignistyp sowie eine Liste der Ereignisse für den Fall, dass der Quelltyp „Parametergruppe“ ist.

Kategorie	Beschreibung
Konfigurationsänderung	Der Parameter %s wurde mit der Methode %s auf %s aktualisiert

Überwachen von Amazon DocumentDB mit CloudWatch

Amazon DocumentDB (mit MongoDB-Kompatibilität) lässt sich in Amazon integrieren CloudWatch sodass Sie Betriebsmetriken für Ihre Cluster sammeln und analysieren können. Sie können diese Metriken mit dem überwachen CloudWatch Konsole, die Amazon DocumentDB-Konsole,AWS Command Line Interface(AWS CLI), oder die CloudWatchAPI.

CloudWatch ermöglicht es Ihnen auch, Alarme einzustellen, sodass Sie benachrichtigt werden können, wenn ein Metrikwert einen von Ihnen angegebenen Schwellenwert überschreitet. Sie können sogar Amazon einrichten CloudWatch Ereignisse, um im Falle eines Verstoßes Abhilfemaßnahmen zu ergreifen. Weitere Informationen zur Verwendung von CloudWatch und Alarme finden Sie imAmazon CloudWatch Dokumentation.

Themen

Amazon DocumentDB-Metriken

Um den Zustand und die Leistung Ihres Amazon DocumentDB-Clusters und Ihrer Amazon DocumentDB-Instances zu überwachen, können Sie die folgenden Metriken in der Amazon DocumentDB-Konsole einsehen.

> **Note**
>
> Die Metriken in den folgenden Tabellen gelten sowohl für instanzbasierte als auch für elastische Cluster.

Nutzung der Ressourcen

Metrik	Beschreibung
`BackupRetentionPeriodStorageUsed`	Die Gesamtmenge des Backup-Speichers in GiB, der zur Unterstützung von verwendet wird point-in-time Wiederherstellungsfunktion innerhalb des Aufbewahrungsfensters von Amazon DocumentDB. Ist in dem von der `TotalBackupStorageBilled` -Metrik gemeldeten Gesamtwert enthalten. Wird für jeden Amazon DocumentDB-Cluster separat berechnet.
`ChangeStreamLogSize`	Die Menge des Speichers, der vom Cluster zum Speichern des Änderungsstream-Protokolls in Megabyte verwendet wird. Dieser Wert ist eine Teilmenge des Gesamtspeichers für den Cluster (`VolumeBytesUsed`) und wirkt sich auf die Kosten des Clusters aus. Preisinformationen für Speicherplatz finden Sie auf derAmazon DocumentDB-Produktseite. Die Protokollgröße des Änderungsstreams ist vom Umfang der Änderungen auf Ihrem Cluster und der Aufbewahrungsdauer des Änderungsstream-Pr

Metrik	Beschreibung
	otokoll abhängig. Weitere Informationen zu Änderungs streams finden Sie unter Change Streams mit Amazon DocumentDB verwenden.
`CPUUtilization`	Prozentsatz der CPU-Nutzung durch eine Instance.
`DatabaseConnections`	Die Anzahl der offenen Verbindungen auf einer Instance, die im Abstand von einer Minute ausgeführt wurde.
`DatabaseConnectionsMax`	Die maximale Anzahl offener Datenbankverbindungen auf einer Instance in einem Zeitraum von einer Minute.
`DatabaseCursors`	Die Anzahl der auf einer Instance geöffneten Cursor, die im Abstand von einer Minute aufgerufen wurden.
`DatabaseCursorsMax`	Die maximale Anzahl offener Cursor auf einer Instanz in einem Zeitraum von einer Minute.
`DatabaseCursorsTimedOut`	Die Anzahl der Cursor, bei denen innerhalb einer Minute ein Timeout aufgetreten ist.
`FreeableMemory`	Die Menge des verfügbaren Arbeitsspeichers (RAM-Speicher) in Bytes.

Metrik	Beschreibung
FreeLocalStorage	Diese Metrik gibt die Menge an Speicher an, der für jede Instance für temporäre Tabellen und Protokolle zur Verfügung steht. Dieser Wert hängt von der Instance-Klasse ab. Sie können die Menge an freiem Speicherplatz für eine Instance erhöhen, indem Sie eine größere Instance-Klasse für Ihre Instance auswählen.
LowMemThrottleQueueDepth	Die Warteschlangentiefe für Anfragen, die aufgrund des geringen verfügbaren Speichers gedrosselt werden, und werden in einer Frequenz von einer Minute abgerufen.
LowMemThrottleMaxQueueDepth	Die maximale Warteschlangentiefe für Anfragen, die aufgrund von zu wenig verfügbarem Speicherplatz innerhalb einer Minute gedrosselt werden.
LowMemNumOperationsThrottled	Die Anzahl der Anfragen, die aufgrund von zu wenig verfügbarem Speicherplatz innerhalb einer Minute gedrosselt werden.

Metrik	Beschreibung
SnapshotStorageUsed	Die Gesamtmenge des Backup-Speichers in GiB, der von allen Snapshots für einen bestimmten Amazon DocumentDB-Cluster außerhalb des Aufbewahrungszeitfensters für Backups verbraucht wird. Ist in dem von der TotalBackupStorageBilled -Metrik gemeldeten Gesamtwert enthalten. Wird für jeden Amazon DocumentDB-Cluster separat berechnet.
SwapUsage	Größe des auf der Instance genutzten Auslagerungsbereichs.
TotalBackupStorageBilled	Die Gesamtmenge an Backup-Speicher in GiB, die Ihnen für einen bestimmten Amazon DocumentDB-Cluster in Rechnung gestellt wird. Umfasst den Sicherungsspeicher gemessen an den Metriken SnapshotStorageUsed und BackupRetentionPeriodStorageUsed . Wird für jeden Amazon DocumentDB-Cluster separat berechnet.

Metrik	Beschreibung	
`TransactionsOpen`	Die Anzahl der offenen Transaktionen auf einer Instance, die im Abstand von einer Minute ausgeführt wurden.	
`TransactionsOpenMax`	Die maximale Anzahl von Transaktionen, die innerhalb einer Minute auf einer Instance geöffnet werden.	
`VolumeBytesUsed`	Die Menge des von Ihrem Cluster verwendeten Speicherplatzes in Byte. Dieser Wert wirkt sich auf die Kosten des Clusters aus. Preisinformationen finden Sie imAmazon DocumentDB-Produktseite.	

Latency

Metrik	Beschreibung	
`DBClusterReplicaLagMaximum`	Die maximale Verzögerung in Millisekunden zwischen der primären Instance und jeder Amazon DocumentDB-Instance im Cluster.	
`DBClusterReplicaLagMinimum`	Die Mindestverzögerung zwischen der Primär-Instance und jeder Replikat-Instance im Cluster, angegeben in Millisekunden.	

Metrik	Beschreibung
DBInstanceReplicaLag	Die Verzögerung in Millisekunden, wenn Updates von der Primär-Instance an eine Replikat-Instance repliziert werden.
ReadLatency	Die durchschnittliche Dauer für einen Festplatten-E/A-Vorgang.
WriteLatency	Die durchschnittliche Dauer einer Datenträger-E/A-Operation in Millisekunden.

Operationen

Metrik	Beschreibung
DocumentsDeleted	Die Anzahl der gelöschten Dokumente in einem Zeitraum von einer Minute.
DocumentsInserted	Die Anzahl der eingefügten Dokumente in einem Zeitraum von einer Minute.
DocumentsReturned	Die Anzahl der zurückgegebenen Dokumente in einem Zeitraum von einer Minute.
DocumentsUpdated	Die Anzahl der aktualisierten Dokumente in einem Zeitraum von einer Minute.

Metrik	Beschreibung
OpcountersCommand	Die Anzahl der Befehle, die in einem Zeitraum von einer Minute ausgegeben wurden.
OpcountersDelete	Die Anzahl der Löschvorgänge, die in einem Zeitraum von einer Minute ausgeführt wurden.
OpcountersGetmore	Die Anzahl der innerhalb einer Minute ausgegebenen Getmores.
OpcountersInsert	Die Anzahl der innerhalb einer Minute ausgeführten Insert-Operationen.
OpcountersQuery	Die Anzahl der Abfragen, die in einem Zeitraum von einer Minute ausgegeben wurden.
OpcountersUpdate	Die Anzahl der Aktualisierungsvorgänge, die in einem Zeitraum von einer Minute ausgeführt wurden.
TransactionsStarted	Die Anzahl der Transaktionen, die innerhalb einer Minute auf einer Instance gestartet wurden.
TransactionsCommitted	Die Anzahl der Transaktionen, die innerhalb einer Minute auf einer Instance festgeschrieben wurden.

Metrik	Beschreibung
TransactionsAborted	Die Anzahl der Transaktionen, die auf einer Instance in einem Zeitraum von einer Minute abgebrochen wurden.
TTLDeletedDocuments	Die Anzahl der Dokumente , die von einem TTLMonitor in einem Zeitraum von einer Minute gelöscht wurden.

Durchsatz

Metrik	Beschreibung
NetworkReceiveThroughput	Der von Clients erhaltene Netzwerkdurchsatz für jede Instance im Cluster, angegeben in Bytes pro Sekunde. Dieser Durchsatz beinhaltet nicht den Netzwerkdatenverkehr zwischen den Instances im Cluster und dem Cluster-Volume.
NetworkThroughput	Die Menge des Netzwerkdurchsatzes in Byte pro Sekunde, der von jeder Instance im Amazon DocumentDB-Cluster sowohl von Clients empfangen als auch an diese übertragen wird. Dieser Durchsatz beinhaltet nicht den Netzwerkdatenverkehr zwischen den

Metrik	Beschreibung	
	Instances im Cluster und dem Cluster-Volume.	
`NetworkTransmitThroughput`	Der an Clients gesendete Netzwerkdurchsatz für jede Instance im Cluster, angegeben in Bytes pro Sekunde. Dieser Durchsatz beinhaltet nicht den Netzwerkdatenverkehr zwischen den Instances im Cluster und dem Cluster-Volume.	
`ReadIOPS`	Durchschnittliche Anzahl der Festplatten-I/O-Lesevorgänge pro Sekunde. Amazon DocumentDB meldet Lese- und Schreib-IOPS getrennt und in Intervallen von einer Minute.	
`ReadThroughput`	Die durchschnittliche Anzahl Byte, die pro Sekunde vom Datenträger gelesen werden.	

Metrik	Beschreibung	
`VolumeReadIOPs`	Die durchschnittliche Anzahl von in Rechnung gestellten I/O-Operationen aus einem Cluster-Volumen, meldet Werte in einem 5-Minuten -Intervall. In Rechnung gestellte Lesevorgänge werden auf Cluster-Volume-Ebene berechnet, aus allen Instances im Cluster zusammengestellt und dann in fünfminütigen Intervall en gemeldet. Der Wert wird anhand des Werts der Metrik für Leseoperationen über einen fünfminütigen Zeitraum berechnet. Sie können die Menge an in Rechnung gestellten Leseoperationen pro Sekunde bestimmen, indem Sie den Wert der Metrik für in Rechnung gestellte Leseoperationen durch 300 Sekunden teilen. Zum Beispiel, wenn`VolumeReadIOPs` gibt 13.686 zurück, dann sind die abgerechneten Lesevorgänge pro Sekunde 45 (13.686/300 = 45,62). In Rechnung gestellte Operationen fallen für Abfragen für nicht im Buffer-	

Metrik	Beschreibung	
	Cache enthaltene Datenbank seiten an, die erst aus dem Speicher geladen werden müssen. Sie sehen evtl. Spitzenwerte in den in Rechnung gestellten Operationen, da Abfrageergebnisse aus dem Speicher gelesen und anschließend in den Buffer-Cache geladen werden.	

Metrik	Beschreibung	
`VolumeWriteIOPs`	Die durchschnittliche Anzahl von in Rechnung gestellte n Schreib-E/A-Operationen aus einem Cluster-Volume, gemeldet in 5-Minuten-Interval len. In Rechnung gestellte Schreibvorgänge werden auf Cluster-Volume-Ebene berechnet, aus allen Instances im Cluster aggregiert und dann in 5-Minuten-Interval len gemeldet. Der Wert wird anhand des Werts der Schreiboperationsmetrik über einen Zeitraum von 5 Minuten berechnet. Sie können die Menge der in Rechnung gestellten Schreiboperationen pro Sekunde ermitteln, indem Sie den Wert der Schreibop erations-Abrechnungsmetrik durch 300 Sekunden teilen. Zum Beispiel, wenn`VolumeWriteIOPs` gibt 13.686 zurück, dann belaufen sich die abgerechneten Schreibvorgänge pro Sekunde auf 45 (13.686/300 = 45,62). Beachten Sie, dass`VolumeRea dIOPs` und`VolumeWri teIOPs` Die Metriken werden von der DocumentDB-	

Metrik	Beschreibung	
	Speicherebene berechnet und umfassen die von der Primär- und Replikatinstanz ausgeführten IOs. Die Daten werden alle 20 bis 30 Minuten aggregiert und dann in Intervallen von 5 Minuten gemeldet, sodass im Zeitraum derselbe Datenpunkt für die Metrik ausgegeben wird. Wenn Sie nach einer Metrik suchen, die mit Ihren Einfügevorgängen über ein Intervall von 1 Minute korreliert, können Sie die WriteIOps -Metrik auf Instanzebene verwenden. Die Metrik ist auf der Registerkarte „Überwachung“ Ihrer primären Amazon DocumentDB-Instance verfügbar.	
`WriteIOPS`	Durchschnittliche Anzahl von Festplatten-I/O-Schreibvorgänge pro Sekunde. Bei Verwendung auf Cluster-Ebene`WriteIOPs` werden für alle Instanzen im Cluster ausgewertet. Lese- und Schreib-IOPS werden separat und in 1-Minuten-Intervallen angegeben.	

Metrik	Beschreibung
`WriteThroughput`	Die durchschnittliche Anzahl von Bytes, die pro Sekunde auf den Datenträger geschrieben werden.

System (System)

Metrik	Beschreibung
`BufferCacheHitRatio`	Der Prozentsatz der vom Buffer-Cache bedienten Anfragen.
`DiskQueueDepth`	die Anzahl der gleichzeitigen Schreibanforderungen auf das verteilte Speichervolume.
`EngineUptime`	Die Gesamtlaufzeit der Instance in Sekunden.
`IndexBufferCacheHitRatio`	Der Prozentsatz der Indexanfragen, die vom Puffercache bedient werden. Möglicherweise stellen Sie unmittelbar nach dem Löschen eines Indexes, einer Sammlung oder einer Datenbank einen Anstieg von mehr als 100% für die Metrik fest. Dies wird nach 60 Sekunden automatisch korrigiert. Diese Einschränkung wird in einem zukünftigen Patch-Update behoben.

T3-Instanz-Metriken

Metrik	Beschreibung
`CPUCreditUsage`	Die Anzahl der CPU-Credits, die während des Messzeitr aums ausgegeben wurden.
`CPUCreditBalance`	Die Anzahl der CPU-Credits, die eine Instance gesammelt hat. Dieser Saldo wird während der Steigerung der CPU-Leistung aufgebrau cht, da die CPU-Guthaben schneller verbraucht als erworben werden.
`CPUSurplusCreditBa lance`	Die Anzahl der überschüs sigen CPU-Guthaben, die zur Aufrechterhaltung der CPU-Leistung ausgegeben wurden, wenn die CPUCreditBalance der Wert ist Null.
`CPUSurplusCreditsC harged`	Die Anzahl der überschüs sigen CPU-Guthaben, die die maximale Anzahl an CPU-Credits übersteig t, die innerhalb von 24 Stunden verdient werden können, weshalb zusätzliche Gebühren anfallen. Weitere Informationen finden Sie unterÜberwachen Sie Ihre CPU-Guthaben.

Wird angezeigt CloudWatch Daten

Sie können Amazon ansehen CloudWatch Daten mit dem CloudWatch Konsole, die Amazon DocumentDB-Konsole,AWS Command Line Interface(AWS CLI), oder die CloudWatch API.

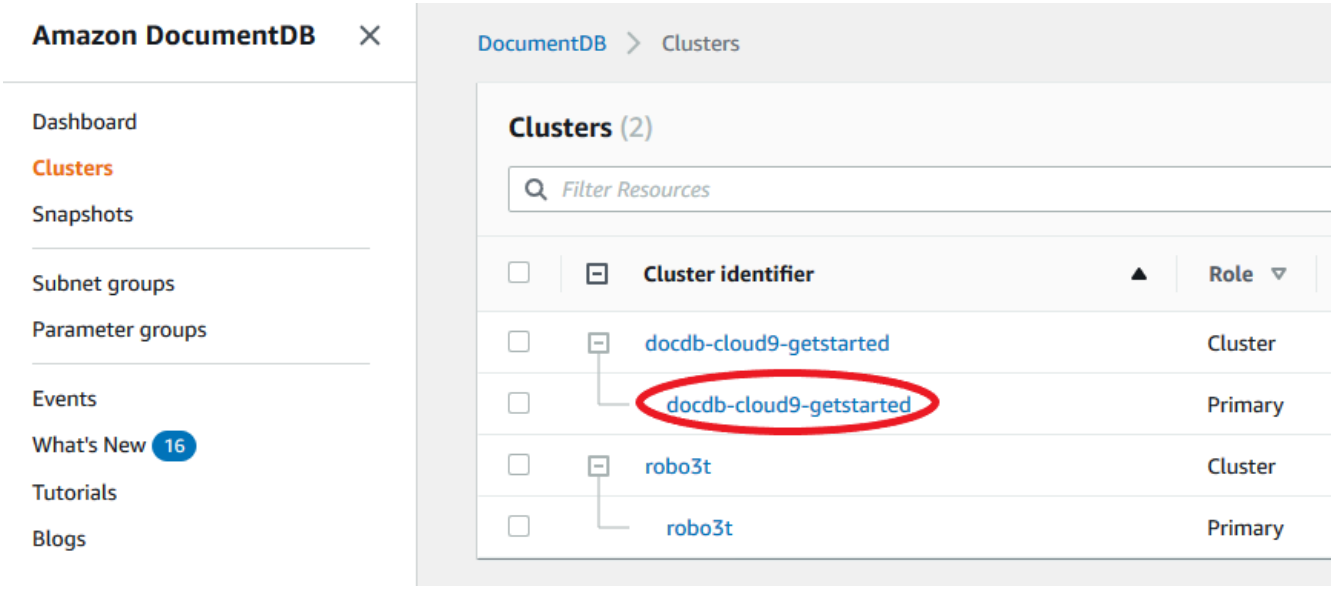
Using the AWS Management Console

Zum Ansehen CloudWatch Führen Sie mithilfe der Amazon DocumentDB Management Console die folgenden Schritte aus.

1. Melden Sie sich beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster).

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

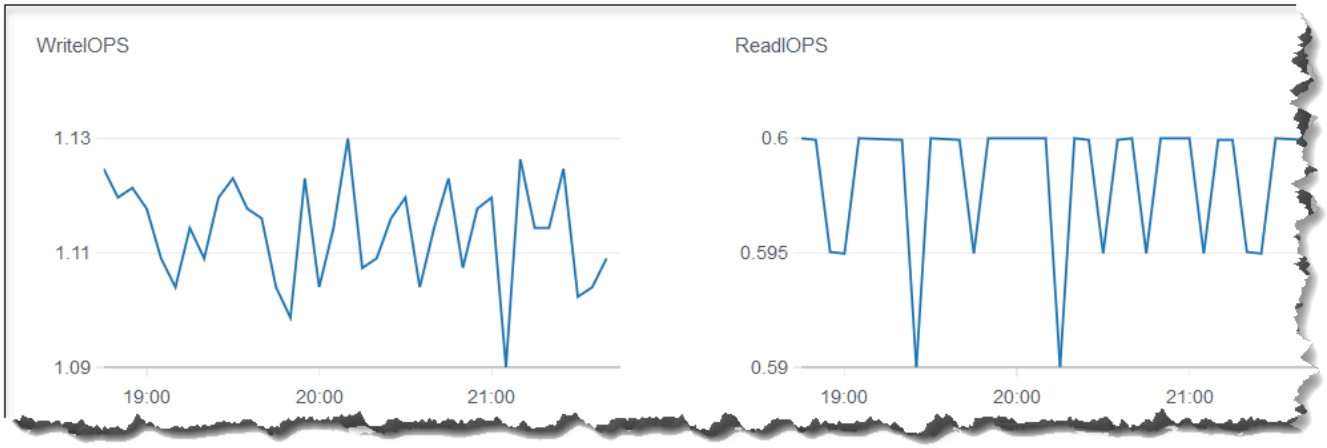
3. Im Cluster-Navigationsfeld sehen Sie die SpalteCluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Wählen Sie aus der Liste der Instanzen den Namen der Instanz aus, für die Sie Metriken benötigen.

5. Wählen Sie auf der daraufhin angezeigten Seite mit der Zusammenfassung der InstanzÜberwachungRegisterkarte, um grafische Darstellungen der Metriken Ihrer Amazon DocumentDB-Instance anzuzeigen. Da für jede Metrik ein Diagramm generiert werden muss, kann es einige Minuten dauern, bisCloudWatchGrafiken zum Auffüllen.

   Die folgende Abbildung zeigt die grafischen Darstellungen von zwei CloudWatch Metriken in der Amazon DocumentDB-Konsole,`WriteIOPS`und`ReadIOPS`.



Using the CloudWatch Management Console

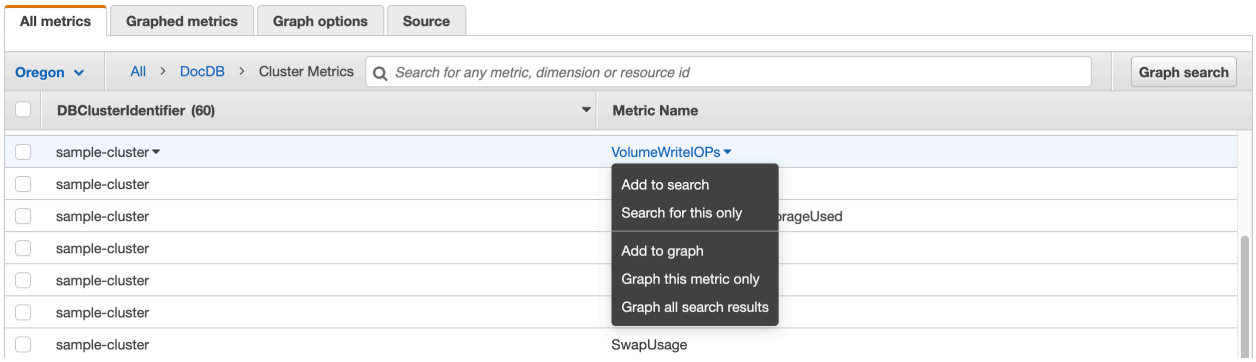
Zum Ansehen CloudWatch Metriken unter Verwendung der CloudWatch Management Console, führen Sie die folgenden Schritte aus.

1. Melden Sie sich beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/cloudwatch.
2. Wählen Sie im Navigationsbereich Metriken aus. Wählen Sie dann aus der Liste der DienstnamenDocDB.
3. Wählen Sie eine metrische Dimension (zum BeispielCluster-Metriken).
4. DieAlle MetrikenAuf der Registerkarte werden alle Metriken für diese Dimension in angezeigtDocDB.

   a. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

   b. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

c. Um nach Metrik zu filtern, bewegen Sie den Mauszeiger über den Metriknamen und wählen Sie den Dropdown-Pfeil neben dem Metriknamen aus. Wählen Sie dannZur Suche hinzufügen, wie in der Abbildung unten gezeigt.



Using the AWS CLI

Zur Ansicht CloudWatch Daten für Amazon DocumentDB verwenden Sie CloudWatch `get-metric-statistics`Operation mit den folgenden Parametern.

Parameter

- **`--namespace`** – Erforderlich. Der Service-Namespace, für den Sie CloudWatch -Metriken anzeigen möchten. Für Amazon DocumentDB muss dies sein`AWS/DocDB`.
- **`--metric-name`** – Erforderlich. Der Name der Metrik, für die Sie Daten benötigen.
- **`--start-time`** – Erforderlich. Der Zeitstempel, der den ersten zurückzugebenden Datenpunkt bestimmt.

  Der angegebene Wert wird eingeschlossen. Die Ergebnisse umfassen Datenpunkte mit dem angegebenen Zeitstempel. Der Zeitstempel muss im ISO 8601 UTC-Format angegeben werden (z. B. 2016-10-03T23:00:00Z).
- **`--end-time`** – Erforderlich. Der Zeitstempel, der den letzten zurückzugebenden Datenpunkt bestimmt.

  Der angegebene Wert wird eingeschlossen. Die Ergebnisse umfassen Datenpunkte mit dem angegebenen Zeitstempel. Der Zeitstempel muss im ISO 8601 UTC-Format angegeben werden (z. B. 2016-10-03T23:00:00Z).
- **`--period`** – Erforderlich. Die Granularität der zurückgegebenen Datenpunkte in Sekunden. Für Metriken mit regulärer Auflösung kann ein Zeitraum gerade mal eine Minute (60 Sekunden) betragen. Der Wert muss ein Vielfaches von 60 sein. Für hochauflösende Metriken, die in

Abständen von weniger als einer Minute erfasst werden, kann der Zeitraum 1, 5, 10, 30, 60 oder ein Vielfaches von 60 betragen.

- **`--dimensions`**— Fakultativ. Wenn die Metrik mehrere Dimensionen enthält, müssen Sie für jede Dimension einen Wert angeben. CloudWatch behandelt jede eindeutige Kombination von Dimensionen als separate Metrik. Wenn eine bestimmte Kombination von Dimensionen nicht veröffentlicht wurde, können Sie keine Metriken dafür abrufen. Sie müssen die gleichen Dimensionen angeben, die bei der Erstellung der Metriken verwendet wurden.
- **`--statistics`**— Fakultativ. Die Metrik-Statistiken abgesehen von den Perzentil-Statistiken. Verwenden Sie für Perzentil-Statistiken `ExtendedStatistics`. Wenn Sie `GetMetricStatistics` aufrufen, müssen Sie `Statistics` oder `ExtendedStatistics` angeben, aber nicht beides.

  Gültige Werte:

  - `SampleCount`
  - `Average`
  - `Sum`
  - `Minimum`
  - `Maximum`
- **`--extended-statistics`**— Fakultativ. Die `percentile`-Statistiken. Geben Sie Werte zwischen p0.0 und p100 an. Wenn Sie `GetMetricStatistics` aufrufen, müssen Sie `Statistics` oder `ExtendedStatistics` angeben, aber nicht beides.
- **`--unit`**— Fakultativ. Die Einheit für eine bestimmte Metrik. Metriken können in mehreren Einheiten angegeben werden. Wenn Sie keine Einheit festlegen, werden alle Einheiten zurückgegeben. Wenn Sie nur eine Einheit angeben, die für die Metrik nicht unterstützt wird, erhalten Sie keine Ergebnisse für den Aufruf.

  Mögliche Werte:

  - `Seconds`
  - `Microseconds`
  - `Milliseconds`
  - `Bytes`
  - `Kilobytes`
  - `Megabytes`

- Gigabytes
- Terabytes
- Bits
- Kilobytes
- Megabits
- Gigabits
- Terabits
- Percent
- Count
- Bytes/Second
- Kilobytes/Second
- Megabytes/Second
- Gigabytes/Second
- Terabytes/Second
- Bits/Second
- Kilobits/Second
- Megabits/Second
- Gigabits/Second
- Terabits/Second
- Count/Second
- None

Example

Im folgenden Beispiel wird die maximale CPUUtilization für einen Zeitraum von 2 Stunden gezeigt, wobei alle 60 Sekunden eine Messung erfolgt.

Für Linux, macOS oder Unix:

```
aws cloudwatch get-metric-statistics \
       --namespace AWS/DocDB \
       --dimensions \
           Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 \
       --metric-name CPUUtilization \
```

```
       --start-time 2019-02-11T05:00:00Z \
       --end-time 2019-02-11T07:00:00Z \
       --period 60 \
       --statistics Maximum
```

Für Windows:

```
aws cloudwatch get-metric-statistics ^
       --namespace AWS/DocDB ^
       --dimensions ^
           Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 ^
       --metric-name CPUUtilization ^
       --start-time 2019-02-11T05:00:00Z ^
       --end-time 2019-02-11T07:00:00Z ^
       --period 60 ^
       --statistics Maximum
```

Die Ausgabe dieser Operation sieht in etwa wie folgt aus.

```
{
       "Label": "CPUUtilization",
       "Datapoints": [
           {
               "Unit": "Percent",
               "Maximum": 4.49152542374361,
               "Timestamp": "2019-02-11T05:51:00Z"
           },
           {
               "Unit": "Percent",
               "Maximum": 4.25000000000485,
               "Timestamp": "2019-02-11T06:44:00Z"
           },

           ********* some output omitted for brevity *********

           {
               "Unit": "Percent",
               "Maximum": 4.33333333331878,
               "Timestamp": "2019-02-11T06:07:00Z"
           }
       ]
   }
```

Abmessungen von Amazon DocumentDB

Die Metriken für Amazon DocumentDB werden anhand der Werte für das Konto oder den Vorgang qualifiziert. Sie können das verwenden CloudWatch Konsole zum Abrufen von Amazon DocumentDB-Daten, die nach einer der Dimensionen in der folgenden Tabelle gefiltert wurden.

Dimension	Beschreibung
`DBClusterIdentifier`	Filtert die Daten, die Sie für einen bestimmten Amazon DocumentDB-Cluster anfordern.
`DBClusterIdentifier, Role`	Filtert die Daten, die Sie für einen bestimmte n Amazon DocumentDB-Cluster anfordern, und aggregiert die Metrik nach Instance-Rolle (WRITER/READER). Sie können beispiels weise Metriken für alle READER-Instances eines Clusters zusammenfassen.
`DBInstanceIdentifier`	Filtert die angeforderten Daten für eine spezifische Datenbank-Instance.

Überwachung von Opcountern

Opcounter-Metriken haben einen Wert ungleich Null (normalerweise ~50) für inaktive Cluster. Das liegt daran, dass Amazon DocumentDB regelmäßige Zustandsprüfungen, interne Abläufe und Aufgaben zur Erfassung von Metriken durchführt.

Überwachen von Datenbankverbindungen

Wenn Sie die Anzahl der Verbindungen mithilfe von Datenbank-Engine-Befehlen wie`db.runCommand( { serverStatus: 1 })`, sehen Sie möglicherweise bis zu 10 Verbindungen mehr als in`DatabaseConnections`durch CloudWatch. Dies liegt daran, dass Amazon DocumentDB regelmäßige Zustandsprüfungen und Aufgaben zur Erfassung von Kennzahlen durchführt, die nicht berücksichtigt werden`DatabaseConnections`.`DatabaseConnections`steht nur für vom Kunden initiierte Verbindungen.

ProtokolDB-API-API-API-API-API-APIAWS CloudTrail

Amazon DocumentDB SNDB-Kompatibilität von Amazon SNDB-Kompatibilität von Amazon SNDB-Benachrichtigungen von Amazon SB-Aktionen von Amazon SB-Aktionen von Amazon DocumentDBAWS DB-API-Diensten bereitstellt.AWS CloudTrail CloudTrail erfasst alleAWS CLI API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-API-Aufrufe Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen S3-Bucket, einschließlich Ereignissen für Amazon DocumentDB. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse Ereignisse auf CloudTrail Ereignisverlauf anzeigen. Mit den von CloudTrail Amazon SB-Informationen gesammelten Informationen können Sie die an Amazon CloudDB-ProtokolDB-kompatibel gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und weitere Angaben bestimmen.

> **Important**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird. Amazon DocumentDB DocumentDB-Konsolen- und API-Aufrufe werden als Aufrufe an die Amazon RDS-API protokolliert.AWS CLI

Weitere Informationen zu AWS CloudTrail finden Sie im AWS CloudTrail-Benutzerhandbuch.

Amazon DocumentDB DocumentDB-Informationen in CloudTrail

CloudTrail wirdAWS-Konto beim Erstellen Ihres Kontos für Sie aktiviert. Wenn eine Aktivität in Amazon DocumentDB SNDB-API-Ereignissen desAWS -Serviceereignissen CloudTrail Ereignisereignissen Ereignisverlauf aufgezeichnet wird. Sie können die neusten Ereignisse in Ihr(em) AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Anzeigen Ereignissen mit CloudTrail Ereignisverlauf Ereignissen mit Ereignisverlauf.

Erstellen SieAWS-Konto einen Trail Trail von Ereignissen für eine kontinuierliche Aufzeichnung von Ereignissen für eine kontinuierliche Aufzeichnung von Ereignissen für eine Aufzeichnung von Ereignissen für Amazon CloudDB-Kompatibilität. Ein Trail ermöglicht CloudTrail die Bereitstellung von Amazon S3-Bucket SN3-Bucket von Amazon S3 S3-Bucket von CloudTrails. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von

Ihnen angegebenen Amazon S3 Bucket bereit. Darüber hinaus können Sie andereAWS -Services konfigurieren, um die in CloudTrail Protokollen Protokollen erfassten Ereignisdaten erfassten Ereignisdaten entsprechend agieren. Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail-Benutzerhandbuch:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfigurieren von Amazon SNS CloudTrail
- Empvon CloudTrail Protokolldateien aus mehreren Regionen Regionen Regionen Regionen Regionen Regionen Regionen
- Empvon CloudTrail Protokolldateien aus mehreren Konten Konten Konten Konten Konten Konten Konten

Jedes Event oder jeder Protokolleintrag enthält Informationen über den Ersteller der Anfrage. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder -Benutzeranmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer gesendet wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Profilierung von Amazon DocumentDB-Vorgängen

Sie können den Profiler in Amazon DocumentDB (mit MongoDB-Kompatibilität) verwenden, um die Ausführungszeit und Details der Operationen zu protokollieren, die auf Ihrem Cluster ausgeführt wurden. Profiler ist für die Überwachung der langsamsten Operationen in Ihrem Cluster nützlich. So können Sie die Leistung einzelner Abfragen und die allgemeine Cluster-Leistung verbessern.

Standardmäßig ist die Profiler-Funktion deaktiviert. Wenn diese Option aktiviert ist, protokolliert der Profiler bei Amazon Vorgänge, die länger dauern als ein vom Kunden definierter Schwellenwert (z. B. 100 ms) CloudWatch Protokolle. Zu den protokollierten Details gehören der Befehl, für den das Profil erstellt wird, die Uhrzeit, die Planübersicht und die Client-Metadaten. Nachdem die Operationen protokolliert wurden CloudWatch Protokolle, die Sie verwenden können CloudWatch Logs Insights

zur Analyse, Überwachung und Archivierung Ihrer Amazon DocumentDB-Profilerstellungsdaten. Häufige Abfragen sind in Abschnitt Häufige Abfragen zu finden.

Wenn aktiviert, verbraucht der Profiler zusätzliche Ressourcen im Cluster. Sie sollten mit einem hohen Schwellenwert (z. B. 500 ms) beginnen und den Wert allmählich senken, um langsame Operationen zu identifizieren. Ein anfänglicher Schwellenwert von 50 ms kann für Anwendungen mit hohem Durchsatz zu Leistungsproblemen im Cluster führen. Der Profiler ist auf Cluster-Ebene aktiviert und funktioniert auf allen Instances und Datenbanken in einem Cluster. Amazon DocumentDB protokolliert Vorgänge bei Amazon CloudWatch Protokolliert nach bestem Wissen.

Amazon DocumentDB erhebt zwar keine zusätzlichen Gebühren für die Aktivierung des Profilers, Ihnen werden jedoch die Standardtarife für die Nutzung von berechnet CloudWatch Protokolle. Für Informationen über CloudWatch Preise für Protokolle finden Sie unterAmazon CloudWatch Preisgestaltung.

Themen

Unterstützte -Vorgänge

Amazon DocumentDB Profiler unterstützt die folgenden Operationen:

- `aggregate`
- `count`
- `delete`
- `distinct`
- `find` (OP_QUERY und Befehl)
- `findAndModify`
- `insert`

- `update`

Einschränkungen

Der Profiler für langsame Abfragen kann nur dann Profiler-Protokolle ausgeben, wenn die gesamte Ergebnismenge der Abfrage in einen Stapel passt und wenn die Ergebnismenge weniger als 16 MB (maximale BSON-Größe) beträgt. Ergebnissätze, die größer als 16 MB sind, werden automatisch in mehrere Batches aufgeteilt.

Die meisten Treiber oder Shells legen möglicherweise eine kleine Standardstapelgröße fest. Sie können die Batchgröße als Teil Ihrer Abfrage angeben. Für die Erfassung langsamer Abfrageprotokolle empfehlen wir eine Batchgröße, die die Größe Ihrer erwarteten Ergebnismenge übersteigt. Wenn Sie sich bezüglich der Größe der Ergebnismenge nicht sicher sind oder wenn sie variiert, können Sie die Batchgröße auch auf eine große Zahl festlegen (z. B. 100.000).

Die Verwendung einer größeren Batchgröße bedeutet jedoch, dass mehr Ergebnisse aus der Datenbank abgerufen werden müssen, bevor eine Antwort an den Client gesendet wird. Bei einigen Abfragen kann dies zu längeren Verzögerungen führen, bis Sie Ergebnisse erhalten. Wenn Sie nicht vorhaben, die gesamte Ergebnismenge zu verwenden, ist es möglich, dass Sie mehr I/Os für die Verarbeitung der Abfrage aufwenden und das Ergebnis dann verwerfen.

Den Amazon DocumentDB Profiler aktivieren

Die Aktivierung des Profilers auf einem Cluster erfolgt in drei Schritten. Stellen Sie sicher, dass alle Schritte abgeschlossen sind, da andernfalls keine Profilerstellungsprotokolle an gesendet werden CloudWatch Protokolle. Profiler wird auf Cluster-Ebene festgelegt und auf allen Datenbanken und Instances des Clusters ausgeführt.

So aktivieren Sie den Profiler in einem Cluster

1. Da Sie eine Standard-Cluster-Parametergruppe nicht ändern können, stellen Sie sicher, dass Sie über eine verfügbare benutzerdefinierte Cluster-Parametergruppe verfügen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen.
2. Ändern Sie mithilfe einer verfügbaren benutzerdefinierten Cluster-Parametergruppe die folgenden Parameter: `profiler`, `profiler_threshold_ms` und `profiler_sampling_rate`. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen ändern.

3. Erstellen oder ändern Sie Ihren Cluster, um die benutzerdefinierte Cluster-Parametergruppe zu verwenden und den Export zu ermöglichen`profiler`protokolliert zu CloudWatch Logs.

In den folgenden Abschnitten wird gezeigt, wie Sie diese Schritte über die AWS Management Console und die AWS Command Line Interface (AWS CLI) implementieren.

Using the AWS Management Console

1. Bevor Sie beginnen, erstellen Sie einen Amazon DocumentDB-Cluster und eine benutzerdefinierte Cluster-Parametergruppe, falls Sie noch keinen haben. Weitere Informationen erhalten Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen und Erstellen eines Amazon DocumentDB-Clusters.
2. Ändern Sie die folgenden Parameter mithilfe einer verfügbaren benutzerdefinierten Cluster-Parametergruppe. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen ändern.

   - `profiler`— Aktiviert oder deaktiviert die Erstellung von Abfrageprofilen. Zugelassene Werte sind `enabled` und `disabled`. Der Standardwert ist `disabled`. Um Profiling zu aktivieren, legen Sie den Wert auf `enabled` fest.
   - `profiler_threshold_ms`— Wann`profiler`ist eingestellt auf`enabled`, alle Befehle, die länger dauern als `profiler-threshold-ms`sind angemeldet CloudWatch. Zugelassene Werte sind `[50-INT_MAX]`. Der Standardwert ist `100`.
   - `profiler_sampling_rate`— Der Anteil der langsamen Operationen, für die ein Profil erstellt oder protokolliert werden sollte. Zugelassene Werte sind `[0.0-1.0]`. Der Standardwert ist `1.0`.
3. Ändern Sie Ihren Cluster so, dass er die benutzerdefinierte Cluster-Parametergruppe verwendet, und legen Sie fest, dass die Profiler-Log-Exporte auf Amazon veröffentlicht werden CloudWatch.

   a. Wählen Sie im Navigationsbereich die Option Clusters (Cluster), um die benutzerdefinierte Parametergruppe zu einem Cluster hinzuzufügen.

   b. Wählen Sie die Schaltfläche links neben dem Namen des Clusters, dem Sie die Parametergruppe zuordnen möchten. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus, um den Cluster zu ändern.

   c. Wählen Sie unter Cluster options (Cluster-Optionen) die benutzerdefinierte Parametergruppe aus dem obigen Schritt aus, um sie Ihrem Cluster hinzuzufügen.

d. UnterExporte protokollieren, wählenProfiler-Protokolleum auf Amazon zu veröffentlichen CloudWatch.

e. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.

f. Nachdem Sie Ihre Änderungen überprüft haben, können Sie diese sofort oder während des nächsten Wartungsfensters unter Scheduling of modifications (Planen von Änderungen) anwenden.

g. Wählen Sie Modify cluster (Cluster ändern) aus, um den Cluster mit der neuen Parametergruppe zu aktualisieren.

Using the AWS CLI

Mit dem folgenden Verfahren wird der Profiler für alle unterstützten Operationen für den Cluster `sample-cluster` aktiviert.

1. Bevor Sie beginnen, stellen Sie sicher, dass Sie über eine benutzerdefinierte Clusterparametergruppe verfügen, indem Sie den folgenden Befehl ausführen und die Ausgabe für eine Clusterparametergruppe überprüfen, die nicht `default` im Namen und die Parametergruppenfamilie `docdb3.6` hat. Wenn Sie über keine nicht standardmäßige Clusterparametergruppe verfügen, siehe Amazon DocumentDB-Cluster-Parametergruppen erstellen.

```
aws docdb describe-db-cluster-parameter-groups \
    --query 'DBClusterParameterGroups[*].
[DBClusterParameterGroupName,DBParameterGroupFamily]'
```

In der folgenden Ausgabe erfüllt nur `sample-parameter-group` beide Kriterien.

```
[
       [
           "default.docdb3.6",
           "docdb3.6"
       ],
       [
           "sample-parameter-group",
           "docdb3.6"
       ]
]
```

2. Ändern Sie mithilfe Ihrer benutzerdefinierten Cluster-Parametergruppe die folgenden Parameter:

   - `profiler`— Aktiviert oder deaktiviert die Erstellung von Abfrageprofilen. Zugelassene Werte sind `enabled` und `disabled`. Der Standardwert ist `disabled`. Um Profiling zu aktivieren, legen Sie den Wert auf `enabled` fest.
   - `profiler_threshold_ms`— Wann`profiler`ist eingestellt auf`enabled`, alle Befehle dauern länger als`profiler -threshold-ms`sind angemeldet CloudWatch. Zugelassene Werte sind `[0-INT_MAX]`. Wenn Sie diesen Wert auf `0` setzen, wird ein Profil für alle unterstützten Operationen erstellt. Der Standardwert ist `100`.
   - `profiler_sampling_rate`— Der Anteil der langsamen Operationen, für die ein Profil erstellt oder protokolliert werden sollte. Zugelassene Werte sind `[0.0-1.0]`. Der Standardwert ist `1.0`.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=enabled,ApplyMethod=immediate \

 ParameterName=profiler_threshold_ms,ParameterValue=100,ApplyMethod=immediate \

 ParameterName=profiler_sampling_rate,ParameterValue=0.5,ApplyMethod=immediate
```

3. Ändern Sie Ihren Amazon DocumentDB-Cluster so, dass er den`sample-parameter-group`benutzerdefinierte Cluster-Parametergruppe aus dem vorherigen Schritt und legt den Parameter fest`--enable-cloudwatch-logs-exports`zu`profiler`.

   Der folgende Code modifiziert den Cluster`sample-cluster`um das zu verwenden`sample-parameter-group`aus dem vorherigen Schritt und fügt hinzu`profiler`zu den aktivierten CloudWatch Protokolliert Exporte.

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --db-cluster-parameter-group-name sample-parameter-group \
       --cloudwatch-logs-export-configuration '{"EnableLogTypes":["profiler"]}'
```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2020-04-07T02:05:12.479Z",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "LatestRestorableTime": "2020-04-08T22:08:59.317Z",
        "Port": 27017,
        "MasterUsername": "test",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
```

```
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-
cluster",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ],
        "DeletionProtection": true
    }
}
```

Den Amazon DocumentDB Profiler deaktivieren

Um den Profiler zu deaktivieren, deaktivieren Sie beide`profiler`Parameter und der Export von`profiler`loggt sich ein zu CloudWatch Logs.

Deaktivieren des Profilers

Sie können den Parameter `profiler` wie folgt entweder mit der AWS Management Console oder AWS CLI deaktivieren.

Using the AWS Management Console

Das folgende Verfahren verwendetAWS Management Consoleum Amazon DocumentDB zu deaktivieren`profiler`.

1. Melden Sie sich an beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus. Wählen Sie dann den Namen der Cluster-Parametergruppe aus, für die Sie den Profiler deaktivieren möchten.
3. Klicken Sie auf der Seite Cluster parameters (Clusterparameter) auf die Schaltfläche links neben dem `profiler`-Parameter und wählen Sie Edit (Bearbeiten).
4. Wählen Sie im Dialogfeld Modify profiler (Profiler ändern) `disabled` in der Liste aus.
5. Wählen Sie Modify Cluster Parameter (Cluster-Parameter ändern).

Using the AWS CLI

Um `profiler` auf einem Cluster mithilfe der AWS CLI zu deaktivieren, ändern Sie den Cluster wie folgt.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=disabled,ApplyMethod=immediate
```

Deaktivieren des Profiler-Protokollexports

Sie können den Export deaktivieren`profiler`loggt sich ein zu CloudWatch Protokolliert mit einer derAWS Management ConsoleoderAWS CLI, wie folgt.

Using the AWS Management Console

Das folgende Verfahren verwendetAWS Management Consoleum den Export von Protokollen durch Amazon DocumentDB zu deaktivieren CloudWatch.

1. Öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Clusters (Cluster). Wählen Sie die Schaltfläche links neben dem Namen des Clusters, für den Sie den Export von Protokollen deaktivieren möchten.
3. Wählen Sie aus dem Menü Actions (Aktionen) die Option Modify (Ändern) aus.
4. Scrollen Sie nach unten zum Abschnitt Log exports (Protokollexporte) und deaktivieren Sie Profiler logs (Profiler-Protokolle).
5. Klicken Sie auf Weiter.
6. Überprüfen Sie Ihre Änderungen und wählen Sie dann aus, wann diese Änderung auf Ihren Cluster angewendet werden soll.
   - Apply during the next scheduled maintenance window (Anwendung während des nächsten geplanten Wartungsfensters)
   - Apply immediately (Sofort anwenden)
7. Wählen Sie Modify Cluster (Cluster bearbeiten).

Using the AWS CLI

Der folgende Code modifiziert den Cluster`sample-cluster` und deaktiviert CloudWatch Profiler-Protokolle.

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Für Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2020-04-08T02:05:17.266Z",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "LatestRestorableTime": "2020-04-09T05:14:44.356Z",
        "Port": 27017,
        "MasterUsername": "test",
        "PreferredBackupWindow": "02:00-02:30",
```

```
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "DeletionProtection": true
    }
}
```

Zugriff auf Ihre Amazon DocumentDB Profiler-Protokolle

Gehen Sie wie folgt vor, um auf Ihre Profilprotokolle bei Amazon zuzugreifen CloudWatch.

1. Öffne das CloudWatch Konsole beihttps://console.aws.amazon.com/cloudwatch/.
2. Stellen Sie sicher, dass Sie sich in derselben Region wie Ihr Amazon DocumentDB-Cluster befinden.
3. Wählen Sie im Navigationsbereich Logs (Logs) aus.

4. Um die Profiler-Protokolle für Ihren Cluster zu finden, wählen Sie in der Liste `/aws/docdb/yourClusterName/profiler` aus.

   Die Profilprotokolle für jede Ihrer Instances sind unter den jeweiligen Instance-Namen verfügbar.

Häufige Abfragen

Im Folgenden finden Sie einige häufige Abfragen, die Sie verwenden können, um Ihre Befehle, für die Profile erstellt wurden, zu analysieren. Für weitere Informationen über CloudWatch Logs Insights finden Sie unterAnalysieren von Protokolldaten mit CloudWatch Logt EinblickeundBeispielabfragen.

Abruf der 10 langsamsten Operationen für eine angegebene Sammlung

```
filter ns="test.foo" | sort millis desc | limit 10
```

Abruf aller Aktualisierungsoperationen für eine Sammlung, die mehr als 60 ms dauerten

```
filter millis > 60 and op = "update"
```

Abruf der 10 langsamsten Operationen im letzten Monat

```
sort millis desc | limit 10
```

Abruf aller Abfragen mit COLLSCAN-Planübersicht

```
filter planSummary="COLLSCAN"
```

Überwachung mit Performance Insights

Performance Insights erweitert die bestehenden Amazon DocumentDB DocumentDB-Überwachungsfunktionen, um die Leistung Ihres Clusters zu veranschaulichen und Sie bei der Analyse aller Probleme zu unterstützen, die sich darauf auswirken. Mit dem Performance Insights Insights-Dashboard können Sie die Datenbanklast visualisieren und die Last nach Wartezeiten, Abfrageanweisungen, Hosts oder Anwendungen filtern.

> **Note**
>
> Performance Insights ist nur für instanzbasierte Amazon DocumentDB 3.6-, 4.0- und 5.0-Cluster verfügbar.

Wie ist es nützlich?

- Datenbankleistung visualisieren — Visualisieren Sie die Last, um festzustellen, wann und wo sich die Last in der Datenbank befindet
- Ermitteln Sie, was die Belastung der Datenbank verursacht — Ermitteln Sie, welche Abfragen, Hosts und Anwendungen zur Belastung Ihrer Instance beitragen
- Ermitteln Sie, wann Ihre Datenbank ausgelastet ist — Vergrößern Sie das Performance Insights Insights-Dashboard, um sich auf bestimmte Ereignisse zu konzentrieren, oder zoomen Sie heraus, um Trends über einen längeren Zeitraum zu betrachten
- Warnung beim Laden der Datenbank — Automatischer Zugriff auf neue Datenbank-Lademetriken, von CloudWatch wo aus Sie die DB-Lademetriken zusammen mit anderen DocumentDB-Metriken überwachen und Warnmeldungen zu diesen einrichten können

Was sind die Einschränkungen von Amazon DocumentDB Performance Insights?

- Performance Insights in der Region AWS GovCloud (US-West) sind noch nicht verfügbar
- Performance Insights for DocumentDB speichert Leistungsdaten für bis zu 7 Tage
- Abfragen, die länger als 1024 KB sind, werden in Performance Insights nicht aggregiert

Themen

Konzepte von Performance Insights

Themen

Durchschnittliche aktive Sitzungen

Datenbanklast (DB-Last) misst den Aktivitätsgrad in Ihrer Datenbank. Die wichtigste Metrik für Performance Insights ist `DB Load`, die jede Sekunde erfasst wird. Die Einheit für die `DBLoad` Metrik ist der Average Active Sessions (AAS) für eine DocumentDB-Instance.

Eine aktive Sitzung ist eine Verbindung, die Arbeit an die DocumentDB-Instanz übermittelt hat und auf eine Antwort wartet. Wenn Sie beispielsweise eine Abfrage an eine DocumentDB-Instanz senden, ist die Datenbanksitzung aktiv, während die Instanz die Abfrage verarbeitet.

Um die Anzahl der durchschnittlich aktiven Sitzungen AAS zu erhalten, ruft Performance Insights die Anzahl der Sitzungen ab, die gleichzeitig eine Abfrage ausführen. Die AAS ist die Gesamtzahl der Sitzungen geteilt durch die Gesamtzahl der Beispiele. Die folgende Tabelle zeigt fünf aufeinanderfolgende Beispiele einer laufenden Abfrage.

Beispiel	Anzahl der Sitzungen, die eine Abfrage ausführen	AAS	Berechnung
1	2	2	2 Sitzungen/1 Beispiel
2	0	1	2 Sitzungen /2 Beispiele
3	4	2	6 Sitzungen /3 Beispiele
4	0	1.5	6 Sitzungen /4 Beispiele

Beispiel	Anzahl der Sitzungen, die eine Abfrage ausführen	AAS	Berechnung
5	4	2	10 Sitzungen /5 Beispiele

Im vorherigen Beispiel beträgt der DB-Load für das Zeitintervall von 1-5 2 AAS. Eine Erhöhung der DB-Last bedeutet, dass im Durchschnitt mehr Sitzungen für die Datenbank ausgeführt werden.

Dimensionen

Die`DB Load` Metrik unterscheidet sich von den anderen Zeitreihenmetriken, da Sie sie in Unterkomponenten aufteilen können, die als Dimensionen bezeichnet werden. Sie können sich Dimensionen als Kategorien für die verschiedenen Merkmale der `DB Load`-Metrik vorstellen. Bei der Diagnose von Leistungsproblemen sind die Dimensionen Wait States und Top Query am nützlichsten.

Wartezustände

Ein Wartestatus bewirkt, dass eine Abfrageanweisung auf das Eintreten eines bestimmten Ereignisses wartet, bevor sie weiter ausgeführt werden kann. Beispielsweise kann eine Abfrageanweisung warten, bis eine gesperrte Ressource entsperrt ist. Durch die Kombination `DB Load` mit Wartezuständen können Sie sich ein vollständiges Bild vom Sitzungsstatus machen. Hier sind verschiedene DocumentDB-Wartezustände:

DocumentDB-Wartestatus	Beschreibung des Wartestatus
Riegeln	Der Latch-Wartestatus tritt auf, wenn die Sitzung darauf wartet, den Pufferpool auszulagern. Häufiges Ein- und Auslagern des Pufferpools kann häufiger vorkommen, wenn das System häufig umfangreiche Abfragen verarbeitet, Sammlungsscans durchführt oder wenn der Pufferpool zu klein ist, um den Arbeitssatz zu verarbeiten.

DocumentDB-Wartestatus	Beschreibung des Wartestatus
CPU	Der CPU-Wartestatus tritt auf, wenn die Sitzung auf die CPU wartet.
CollectionLock	Der CollectionLock Wartestatus tritt ein, wenn die Sitzung darauf wartet, eine Sperre für die Sammlung zu erlangen. Diese Ereignisse treten auf, wenn DDL-Operationen für die Sammlung ausgeführt werden.
DocumentLock	Der DocumentLock Wartestatus tritt ein, wenn die Sitzung darauf wartet, ein Dokument zu sperren. Eine hohe Anzahl gleichzeitiger Schreibvorgänge auf dasselbe Dokument führt zu mehr DocumentLock Wartezuständen in diesem Dokument.
SystemLock	Der SystemLock Wartestatus tritt auf, wenn die Sitzung auf dem System wartet. Dies kann der Fall sein, wenn häufig Abfragen mit langer Laufzeit, lang andauernde Transaktionen oder eine hohe Parallelität im System auftreten.
IO	Der IO-Wartestatus tritt auf, wenn die Sitzung auf den Abschluss der IO wartet.
BufferLock	Der BufferLock Wartestatus tritt ein, wenn die Sitzung darauf wartet, eine Sperre für eine gemeinsam genutzte Seite im Puffer zu erlangen. BufferLockWartezustände können verlängert werden, wenn andere Prozesse die Cursor auf den angeforderten Seiten offen halten.

DocumentDB-Wartestatus	Beschreibung des Wartestatus
LowMemThrottle	Der LowMemThrottle Wartestatus tritt auf, wenn die Sitzung aufgrund einer hohen Speicherauslastung auf der Amazon DocumentDB DocumentDB-Instance wartet. Wenn dieser Status für eine lange Zeit andauert, sollten Sie erwägen, die Instance zu skalieren, um zusätzlichen Speicher bereitzustellen. Weitere Informationen finden Sie unter Resource Governor.
BackgroundActivity	Der BackgroundActivity Wartestatus tritt auf, wenn die Sitzung auf interne Systemprozesse wartet.
Sonstige	Der Wartestatus Andere ist ein interner Wartestatus. Wenn dieser Status über einen längeren Zeitraum andauert, sollten Sie erwägen, diese Abfrage zu beenden. Weitere Informationen finden Sie unter Wie finde und beende ich Abfragen mit langer Laufzeit oder blockierter Ausführung?

Die häufigsten Abfragen

Während bei Wartezuständen Engpässe auftreten, zeigen Abfragen am häufigsten, welche Abfragen am meisten zur Datenbanklast beitragen. Beispielsweise könnten derzeit viele Abfragen gleichzeitig in der Datenbank ausgeführt werden, aber eine einzelne Abfrage könnte 99 % der DB-Last verbrauchen. In diesem Fall könnte die hohe Belastung auf ein Problem mit der Abfrage hinweisen.

Max. vCPU

Das Diagramm Datenbanklast im Dashboard dient zum Erfassen, Aggregieren und Anzeigen von Sitzungsinformationen. Um zu sehen, ob aktive Sitzungen die maximale CPU überschreiten, sehen Sie sich ihre Beziehung zur Max vCPU-Linie an. Der Wert Max vCPU wird durch die Anzahl der vCPU-Kerne (virtuelle CPU) für Ihre DocumentDB-Instance bestimmt.

Wenn die DB-Last häufig über der Max vCPU-Linie liegt und der primäre Wartezustand „CPU“ lautet, ist die CPU überlastet. In diesem Fall möchten Sie möglicherweise die Verbindungen zur Instanz drosseln, alle Abfragen mit hoher CPU-Last optimieren oder eine größere Instance-Klasse in Betracht ziehen. Hohe und konsistente Instances von Wartezuständen deuten darauf hin, dass es möglicherweise Engpässe oder Probleme mit Ressourcenkonflikten gibt, die behoben werden müssen. Dies kann auch dann zutreffen, wenn die DB-Last die mit Max vCPU definierte Linie nicht überschreitet.

Aktivieren und Deaktivieren von Performance Insights

Um Performance Insights zu nutzen, aktivieren Sie es auf Ihrer DB-Instance. Sie können sie bei Bedarf später deaktivieren. Das Aktivieren und Deaktivieren von Performance Insights führt nicht zu Ausfallzeiten, einem Neustart oder einem Failover.

Der Performance Insights-Agent verbraucht eine begrenzte Menge an CPU und Arbeitsspeicher auf dem DB-Host. Wenn die DB-Last hoch ist, begrenzt der Agent die Auswirkungen auf die Leistung, indem Daten seltener erfasst werden.

Aktivieren von Performance Insights beim Erstellen eines Clusters

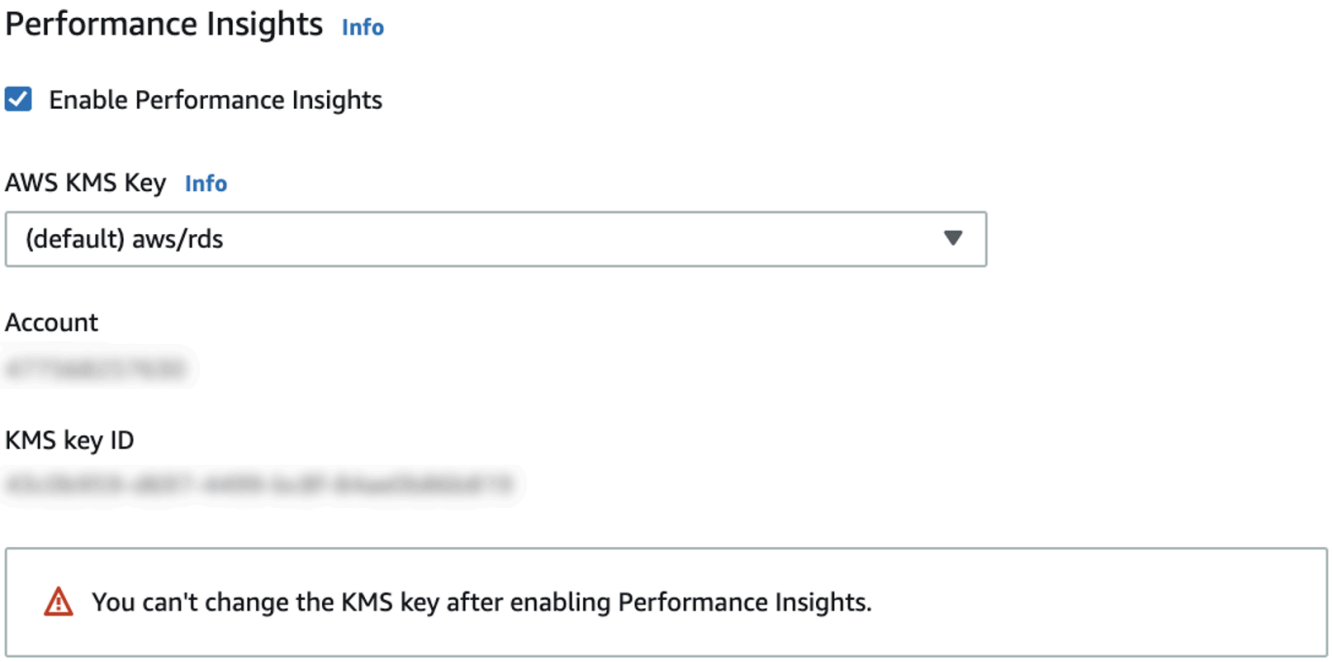
In der Konsole können Sie Performance Insights aktivieren oder deaktivieren, wenn Sie eine neue DB-Instance erstellen oder ändern.

Verwendung der AWS Management Console

In der Konsole können Sie Performance Insights aktivieren, wenn Sie einen DocumentDB-Cluster erstellen. Wenn Sie einen neuen DocumentDB-Cluster erstellen, aktivieren Sie Performance Insights, indem Sie im Abschnitt Performance Insights die Option Performance Insights aktivieren auswählen.

Anweisungen für die Konsole

1. Um einen Cluster zu erstellen, folgen Sie den Anweisungen unter Erstellen eines Amazon DocumentDB-Clusters.
2. Wählen Sie im Abschnitt Performance Insights die Option Performance Insights aktivieren aus.



> **Note**
>
> Die Aufbewahrungsfrist für Performance Insights Insights-Daten beträgt sieben Tage.

AWS KMSSchlüssel — Geben Sie Ihren AWS KMS-Schlüssel an. Performance Insights verschlüsselt alle potentiell sensiblen Daten mit Ihrem eigenen AWS KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Konfiguration einer AWS AWS KMS Richtlinie für Performance Insights.

Aktivierung und Deaktivierung beim Ändern einer Instanz

Sie können eine DB-Instance ändern, um Performance Insights mithilfe der Konsole oder zu aktivieren oder zu deaktivierenAWS CLI.

Using the AWS Management Console

Anweisungen für die Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie Clusters (Cluster) aus.

3. Wählen Sie eine DB-Instance aus und wählen Sie Ändern.
4. Wählen Sie im Abschnitt Performance Insights entweder Performance Insights aktivieren oder Performance Insights deaktivieren aus.

   > **Note**
   >
   > Wenn Sie Enable Performance Insights wählen, können Sie Ihren AWS AWS KMS Schlüssel angeben. Performance Insights verschlüsselt alle potentiell sensiblen Daten mit Ihrem eigenen AWS KMS-Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Daten im Ruhezustand verschlüsseln.

5. Klicken Sie auf Weiter.
6. Wählen Sie für Einplanung von Änderungen die Option Sofort anwenden aus. Wenn Sie während des nächsten geplanten Wartungsfensters Anwenden wählen, ignoriert Ihre Instance diese Einstellung und aktiviert Performance Insights sofort.
7. Wählen Sie Modify instance (Instance ändern).

Using the AWS CLI

Wenn Sie die `modify-db-instance` AWS AWS CLI Befehle `create-db-instance` oder verwenden, können Sie Performance Insights durch Angabe aktivieren oder deaktivieren`--enable-performance-insights`, indem Sie angeben`--no-enable-performance-insights`.

Das folgende Verfahren beschreibt, wie Sie Performance Insights für eine DB-Instance mithilfe von aktivieren oder deaktivieren AWSAWS CLI.

AWSAWS CLIAnweisungen

Rufen Sie den `modify-db-instance` AWS AWS CLI Befehl auf und geben Sie die folgenden Werte an:

- `--db-instance-identifer`— Der Name der DB-Instance
- `--enable-performance-insights` zum Aktivieren oder `--no-enable-performance-insights` zum Deaktivieren

Example

Das folgende Beispiel aktiviert Performance Insights für`sample-db-instance`:

For Linux, macOS, or Unix:

```
aws docdb modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights
```

For Windows:

```
aws docdb modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
    --enable-performance-insights
```

Konfigurieren von Zugriffsrichtlinien für Performance Insights

Um auf Performance Insights zugreifen zu können, müssen Sie über die entsprechenden Berechtigungen von AWS Identity and Access Management (IAM) verfügen. Sie haben folgende Möglichkeiten, Zugriff zu gewähren:

- Fügen Sie die verwaltete Richtlinie `AmazonRDSPerformanceInsightsReadOnly` an einen Berechtigungssatz oder eine Rolle an.
- Erstellen Sie eine benutzerdefinierte IAM-Richtlinie und fügen Sie diese an einen Berechtigungssatz oder eine Rolle an.

Wenn Sie bei der Aktivierung von Performance Insights einen vom Kunden verwalteten Schlüssel angegeben haben, stellen Sie außerdem sicher, dass die Benutzer in Ihrem Konto über die Berechtigungen `kms:Decrypt` und `kms:GenerateDataKey` für den KMS-Schlüssel verfügen.

> **Note**
>
> Für die encryption-at-rest Verwaltung von AWS KMS Schlüsseln und Sicherheitsgruppen nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird.

Anhängen der PerformanceInsightsReadOnly AmazonRDS-Richtlinie an einen IAM-Prinzipal

`AmazonRDSPerformanceInsightsReadOnly`ist eine AWS verwaltete Richtlinie, die Zugriff auf alle schreibgeschützten Operationen der Amazon DocumentDB Performance Insights-API gewährt. Derzeit sind alle Operationen in dieser API schreibgeschützt. Wenn Sie `AmazonRDSPerformanceInsightsReadOnly` an einen Berechtigungssatz oder eine Rolle anfügen, kann der Empfänger Performance Insights mit anderen Konsolenfunktionen verwenden.

Erstellen einer benutzerdefinierten IAM-Richtlinie für Performance Insights

Für Benutzer, die nicht über die `AmazonRDSPerformanceInsightsReadOnly`-Richtlinien verfügen, können Sie den Zugriff auf Performance Insights gewähren, indem Sie eine benutzerverwaltete IAM-Richtlinie erstellen oder ändern. Wenn Sie die Richtlinie an einen Berechtigungssatz oder eine Rolle anhängen, kann der Empfänger Performance Insights verwenden.

Erstellen eine benutzerdefinierten Richtlinie

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies aus.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie auf der Seite Create Policy (Richtlinie erstellen) die Registerkarte „JSON“ aus.
5. Kopieren Sie den folgenden Text und ersetzen Sie *`us-east-1`* mit dem Namen Ihrer AWS-Region und *`111122223333`* mit Ihrer Kundenkontonummer.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rds:DescribeDBInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "rds:DescribeDBClusters",
            "Resource": "*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": "pi:DescribeDimensionKeys",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetDimensionKeyDetails",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetResourceMetadata",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetResourceMetrics",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:ListAvailableResourceDimensions",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:ListAvailableResourceMetrics",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        }
    ]
}
```

6. Wählen Sie Review policy (Richtlinie prüfen).
7. Geben Sie einen Namen und optional eine Beschreibung für die Richtlinie an und wählen Sie dann Create policy (Richtlinie erstellen) aus.

Sie können die Richtlinie nun an einen Berechtigungssatz oder eine Rolle anfügen. Das folgende Verfahren setzt voraus, dass Sie für diesen Zweck bereits einen Benutzer zur Verfügung haben.

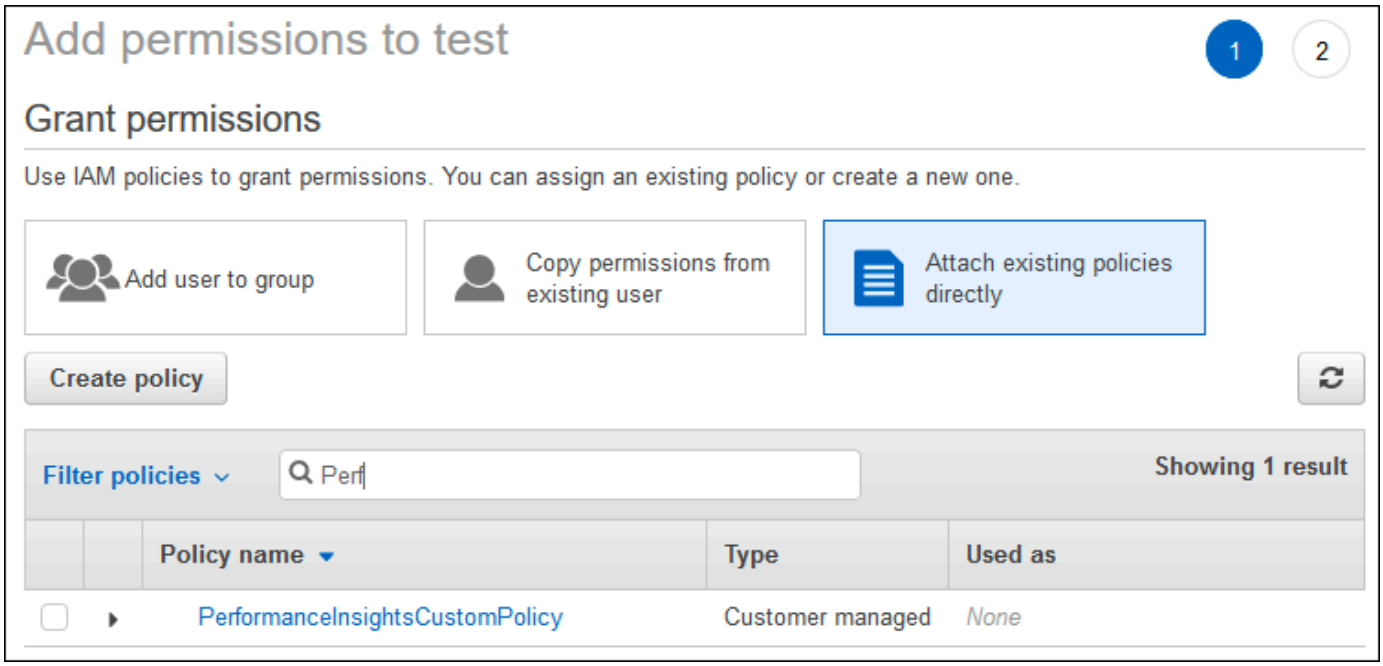
So fügen Sie die Richtlinie an einen Benutzer an

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.

2. Klicken Sie im Navigationsbereich auf Users (Benutzer).
3. Wählen Sie einen vorhandenen Benutzer aus der Liste aus.

> **Important**
>
> Um Performance Insights verwenden zu können, stellen Sie sicher, dass Sie zusätzlich zur benutzerdefinierten Richtlinie Zugriff auf Amazon DocumentDB haben. Die ReadOnlyAccess vordefinierte AmazonDocDB-Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf Amazon DoCDB. Weitere Informationen finden Sie unter Zugriff mithilfe von Richtlinien verwalten.

4. Wählen Sie auf der Seite Summary (Übersicht) die Option Add permissions (Berechtigungen hinzufügen) aus.
5. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen). Geben Sie in Suche die ersten Zeichen Ihres Richtliniennamens ein, wie nachfolgend gezeigt.



6. Wählen Sie Ihre Richtlinie und wählen Sie anschließend Nächster Schritt: Prüfen.
7. Wählen Sie Add permissions (Berechtigungen hinzufügen) aus.

Konfigurieren einer AWS KMS-Richtlinie für Performance Insights

Performance Insights verwendet eine AWS KMS key zur Verschlüsselung sensibler Daten. Wenn Sie Performance Insights über die API oder die Konsole aktivieren, haben Sie folgende Möglichkeiten:

- Wählen Sie den Standardwert Von AWS verwalteter Schlüssel aus.

  Amazon DocumentDB verwendet die Von AWS verwalteter Schlüssel für Ihre neue DB-Instance. Amazon DocumentDB erstellt eine Von AWS verwalteter Schlüssel für Ihr AWS Konto. Ihr AWS Konto hat Von AWS verwalteter Schlüssel für Amazon DocumentDB für jede AWS Region ein anderes.
- Wählen Sie einen kundenverwalteten Schlüssel.

  Wenn Sie einen vom Kunden verwalteten Schlüssel angeben, benötigen Benutzer in Ihrem Konto, die die Performance Insights API aufrufen, die Berechtigungen `kms:Decrypt` und `kms:GenerateDataKey` für den KMS-Schlüssel. Sie können diese Berechtigungen über IAM-Richtlinien konfigurieren. Wir empfehlen jedoch, dass Sie diese Berechtigungen über Ihre KMS-Schlüsselrichtlinie verwalten. Weitere Informationen finden Sie unter Verwenden von Schlüsselrichtlinien in AWS-KMS.

Example

Das folgende Beispiel für eine Schlüsselrichtlinie zeigt, wie Sie Ihrer KMS-Schlüsselrichtlinie Anweisungen hinzufügen können. Diese Anweisungen erlaubt den Zugriff auf Performance Insights. Je nachdem, wie Sie das verwendenAWS KMS, möchten Sie möglicherweise einige Einschränkungen ändern. Bevor Sie Ihrer Richtlinie Anweisungen hinzufügen, entfernen Sie alle Kommentare.

```
{
 "Version" : "2012-10-17",
 "Id" : "your-policy",
 "Statement" : [ {
    //This represents a statement that currently exists in your policy.
 }
 ....,
 //Starting here, add new statement to your policy for Performance Insights.
 //We recommend that you add one new statement for every RDS/DocumentDB instance
 {
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            //One or more principals allowed to access Performance Insights
            "arn:aws:iam::444455556666:role/Role1"
        ]
```

```
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" :{
        "StringEquals" : {
            //Restrict access to only RDS APIs (including Performance Insights).
            //Replace *region* with your AWS Region.
            //For example, specify us-west-2.
            "kms:ViaService" : "rds.*region*.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
            //Restrict access to only data encrypted by Performance Insights.
            "kms:EncryptionContext:aws:pi:service": "rds",
            "kms:EncryptionContext:service": "pi",

            //Restrict access to a specific DocDB instance.
            //The value is a DbiResourceId.
           "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
        }
    }
}
```

Analyse der Metriken mit dem Performance Insights-Dashboard

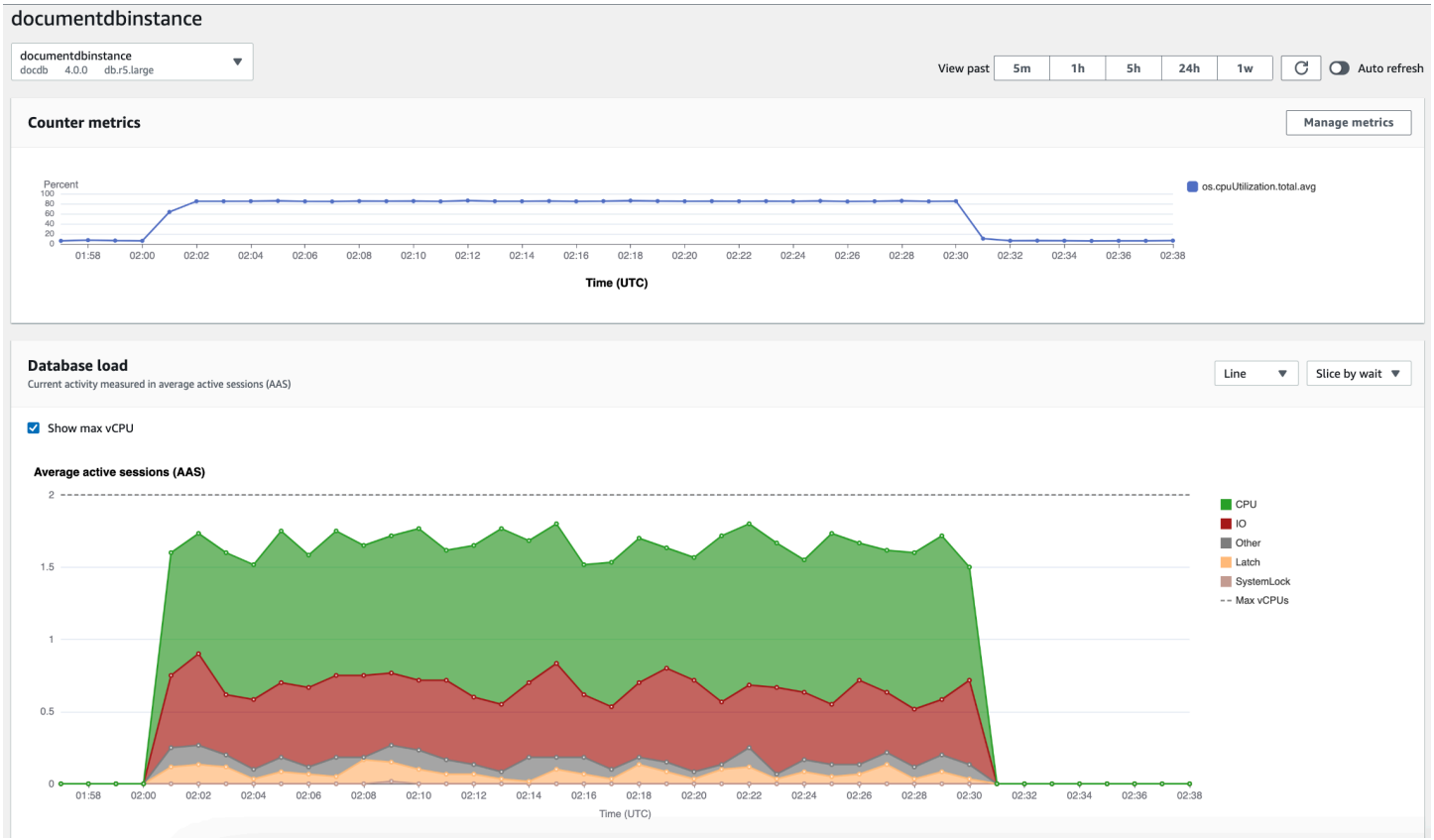
Das Dashboard von Performance Insights enthält Informationen zur Datenbank-Performance, die Sie bei der Analyse und Behebung von Performance-Problemen unterstützen. Auf der Hauptseite des Dashboards können Sie Informationen zur Datenbanklast (DB-Load) einsehen. Sie können die Datenbanklast nach Dimensionen wie Wartestatus oder Abfrage „aufteilen“.

Themen

- Überblick über Performance Insights
- Öffnen des Performance Insights-Dashboards
- Analysieren der Datenbanklast anhand von Wartezuständen
- Überblick über die Registerkarte „Häufigste Abfragen“
- Das Diagramm zum Laden der Datenbank vergrößern

Überblick über Performance Insights

Das Dashboard ist die einfachste Möglichkeit, mit Performance Insights zu interagieren. Das folgende Beispiel zeigt das Dashboard für eine Amazon DocumentDB DocumentDB-Instance. Standardmäßig zeigt das Dashboard von Performance Insights die Daten der letzten Stunde an.



Das Dashboard ist in folgende Teile gegliedert:

1. Zählermetriken — Zeigt Daten für bestimmte Leistungsindikatormetriken an.
2. Datenbanklast — Zeigt, wie die DB-Last im Vergleich zur DB-Instance-Kapazität abschneidet, wie sie in der Zeile Max vCPU dargestellt wird.
3. Top-Dimensionen — Zeigt die wichtigsten Dimensionen an, die zur DB-Auslastung beitragen. Zu diesen Dimensionen gehören `waits`,`queries`,`hosts`,`databases`, und`applications`.

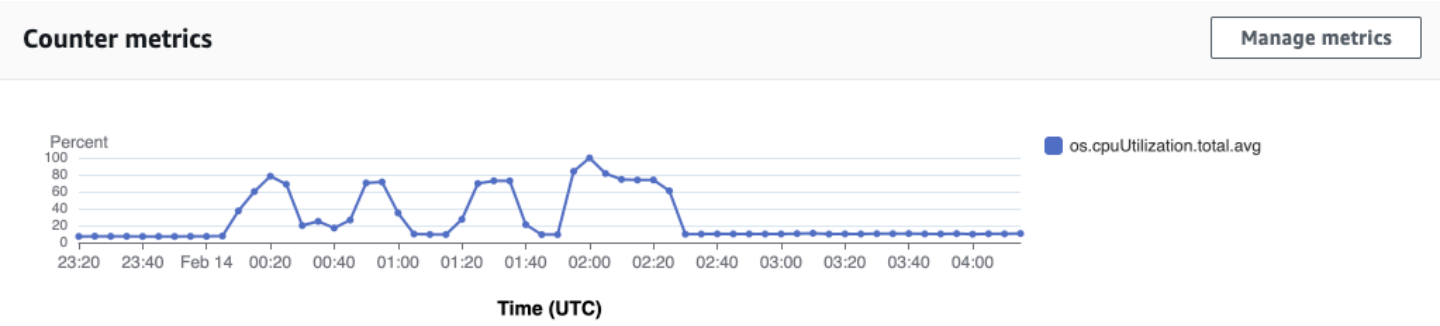
Themen

- Zählermetriken-Diagramm
- Datenbank-Ladediagramm
- Dimensionen pro Tabelle

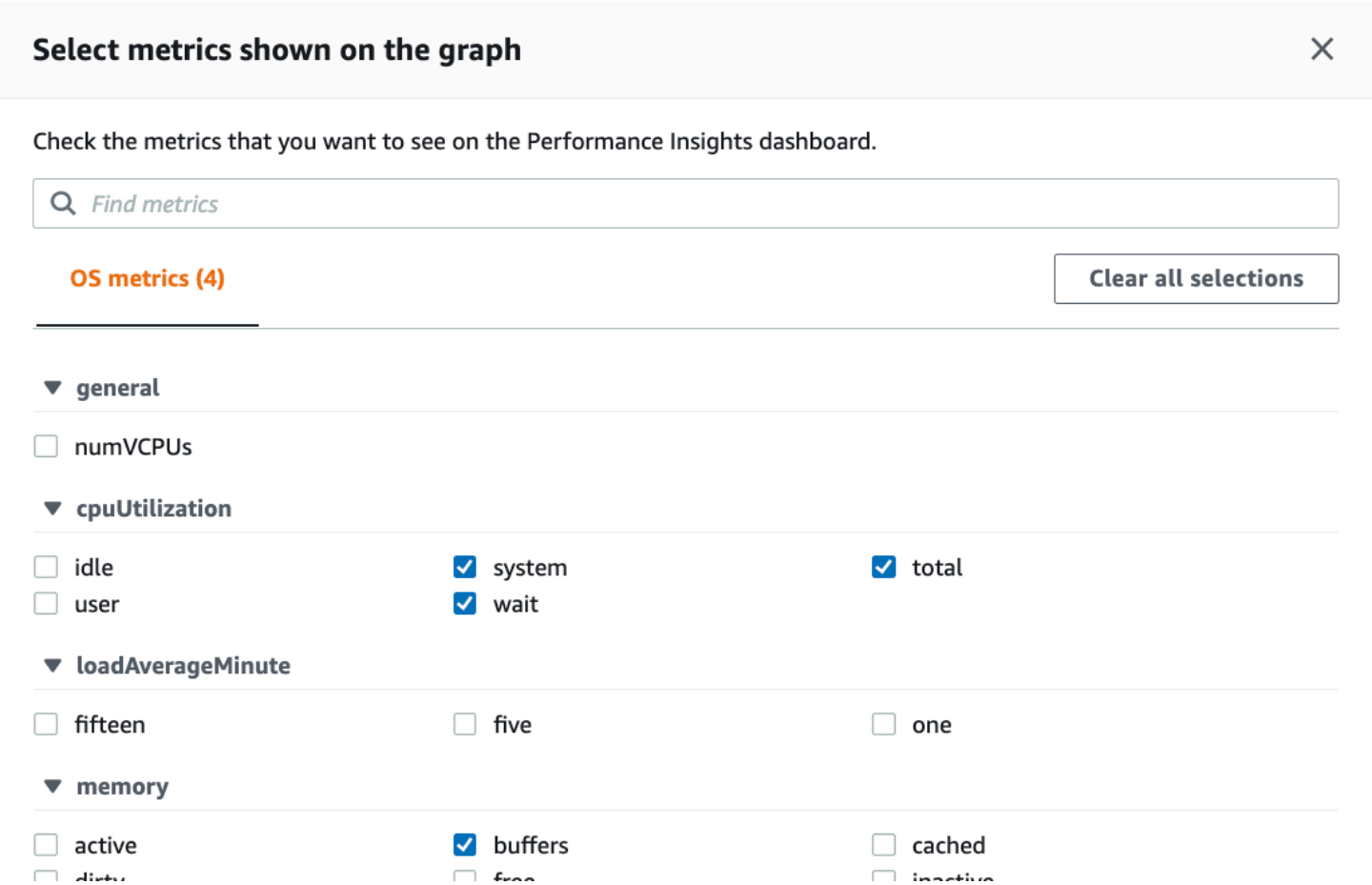
Zählermetriken-Diagramm

Mithilfe von Zählermetriken können Sie das Performance Insights-Dashboard anpassen und bis zu 10 weitere Diagramme aufnehmen. Diese Grafiken zeigen eine Auswahl von Dutzenden von Betriebssystemmetriken. Diese Informationen können mit der Datenbanklast korreliert werden, um Performance-Probleme zu identifizieren und zu analysieren.

Das Counter Metrics (Zählermetriken)-Diagramm enthält Daten zu Leistungsindikatoren.



Um die Leistungsindikatoren zu ändern, wählen Sie Metriken verwalten aus. Sie können mehrere Betriebssystemmetriken auswählen, wie im folgenden Screenshot gezeigt. Um Details für jede Metrik anzuzeigen, bewegen Sie den Mauszeiger über den Metriknamen.



Datenbank-Ladediagramm

Das Diagramm zum Laden der Datenbank zeigt, wie die Datenbankaktivität im Vergleich zur Instanzkapazität abschneidet, wie sie in der Zeile Max. vCPU dargestellt wird. Standardmäßig stellt das gestapelte Liniendiagramm die DB-Last als durchschnittliche aktive Sitzungen pro Zeiteinheit dar. Die DB-Last wird nach Wartestatus aufgeteilt (gruppiert).

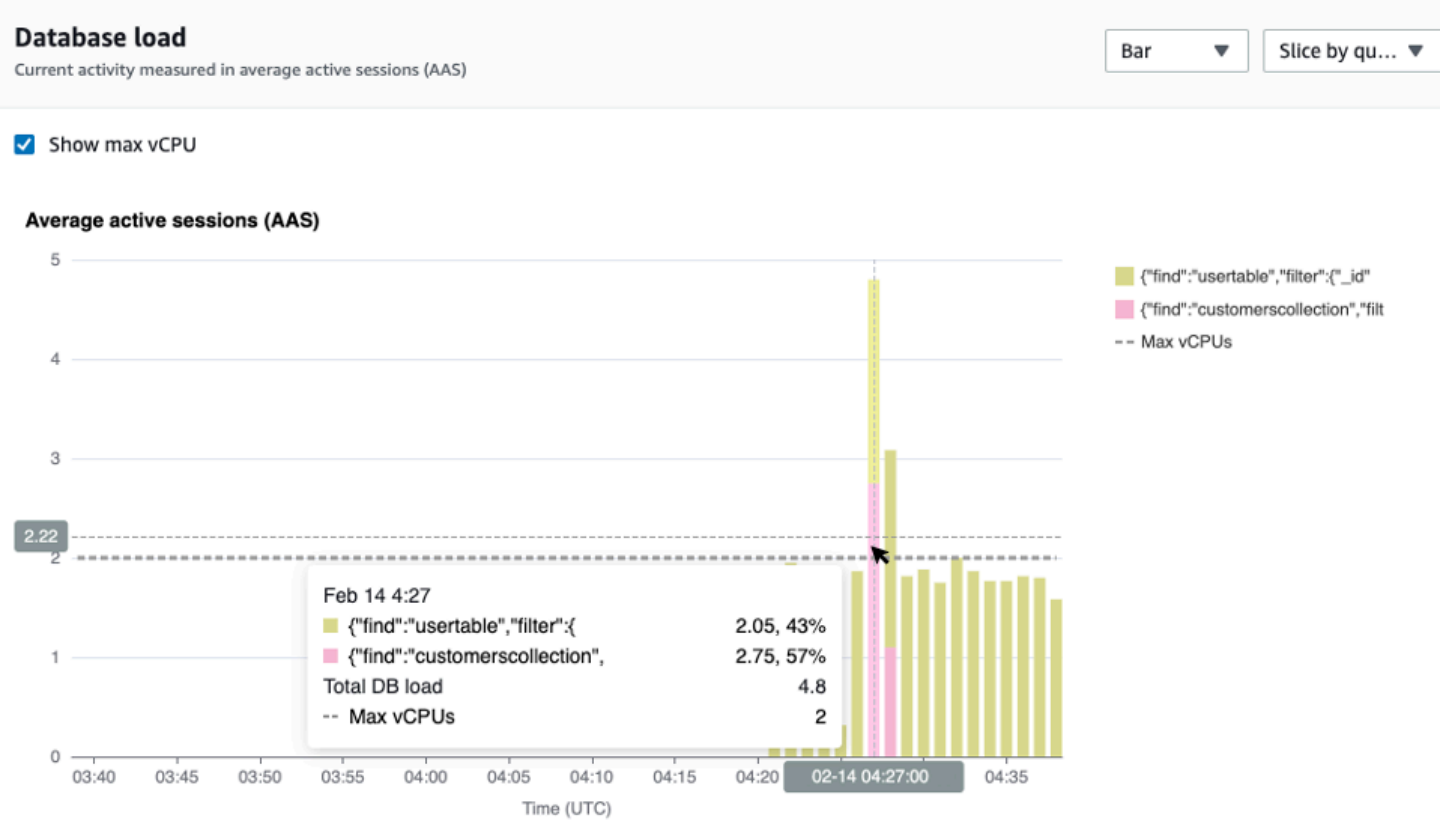
DB-Last aufgeteilt nach Dimensionen

Sie können die Last als aktive Sitzungen anzeigen, die nach unterstützten Dimensionen gruppiert sind. Die folgende Abbildung zeigt die Abmessungen für die Amazon DocumentDB DocumentDB-Instance.

DB-Ladedetails für ein Dimensionselement

Um Details zu einem DB-Lastelement innerhalb einer Dimension anzuzeigen, bewegen Sie den Mauszeiger über den Elementnamen. Die folgende Abbildung zeigt Details für eine Abfrageanweisung.

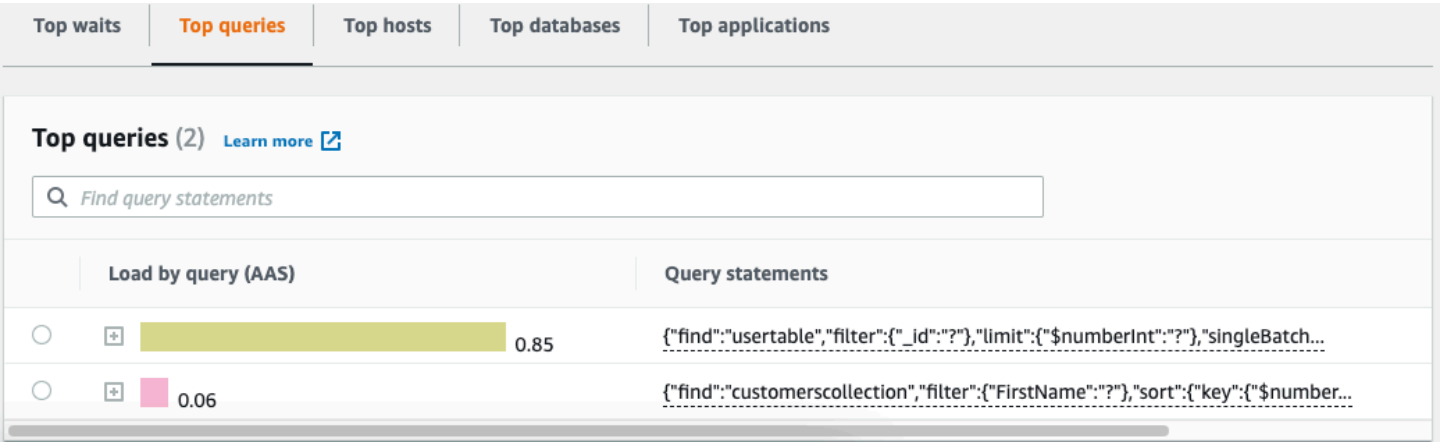
Um Details zu einem Element für den ausgewählten Zeitraum in der Legende anzuzeigen, bewegen Sie den Mauszeiger über dieses Element.



Dimensionen pro Tabelle

In der Tabelle mit den obersten Dimensionen wird die Datenbanklast nach verschiedenen Dimensionen aufgeteilt. Eine Dimension ist eine Kategorie oder „Aufteilung“ für verschiedene Merkmale der DB-Last. Wenn es sich bei der Dimension um eine Abfrage handelt, werden in den häufigsten Abfragen die Abfrageanweisungen angezeigt, die am meisten zur DB-Auslastung beitragen.

Wählen Sie eine der folgenden Dimensionsregisterkarten.



Die folgende Tabelle enthält eine kurze Beschreibung der einzelnen Registerkarten.

Tab	Beschreibung
Top waits (Top-Wartereignis)	Das Ereignis, auf das das Datenbank-Backend wartet
Die häufigsten	**Die** Abfrageanweisungen, Abfragen, die derzeit ausgeführt werden
Top hosts (Top-Hosts)	Die Host-IP und der Port des verbundenen Clients
Top databases	Der Name

Beschreibung
Der Top-Datenbanken (Top-Datenbanken) mit der der Client verbunden ist
Der Top-Anwendungen (Top-Anwendungen) mit der Datenbank verbunden ist

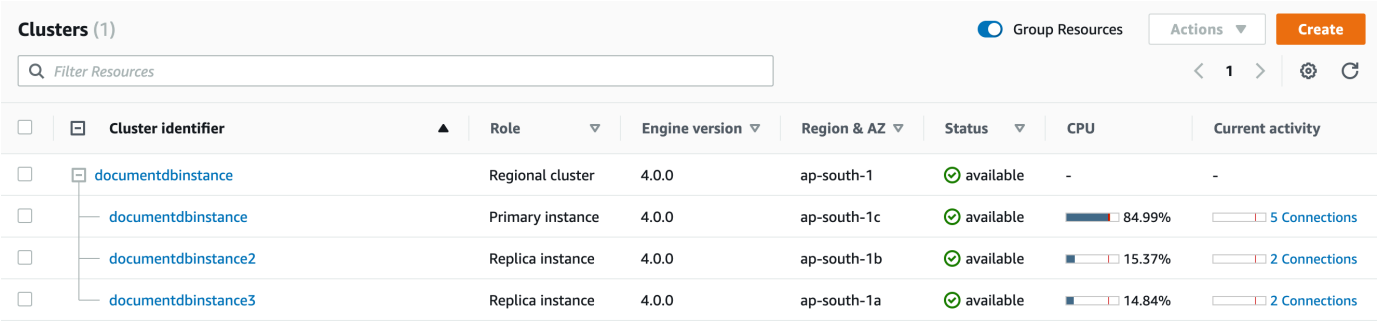
Informationen zum Analysieren von Abfragen mithilfe der Registerkarte „Häufig gestellte Abfragen“ finden Sie unter Überblick über die Registerkarte „Häufigste Abfragen“.

Öffnen des Performance Insights-Dashboards

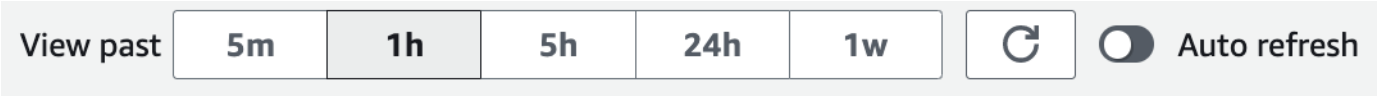
Gehen Sie wie folgt vor, um das Performance Insights Insights-Dashboard in der AWS Management Console anzuzeigen:

1. Öffnen Sie die Performance Insights Insights-Konsole unter https://console.aws.amazon.com/docdb/.
2. Wählen Sie eine DB-Instance aus. Das Performance Insights Insights-Dashboard wird für diese Amazon DocumentDB DocumentDB-Instance angezeigt.

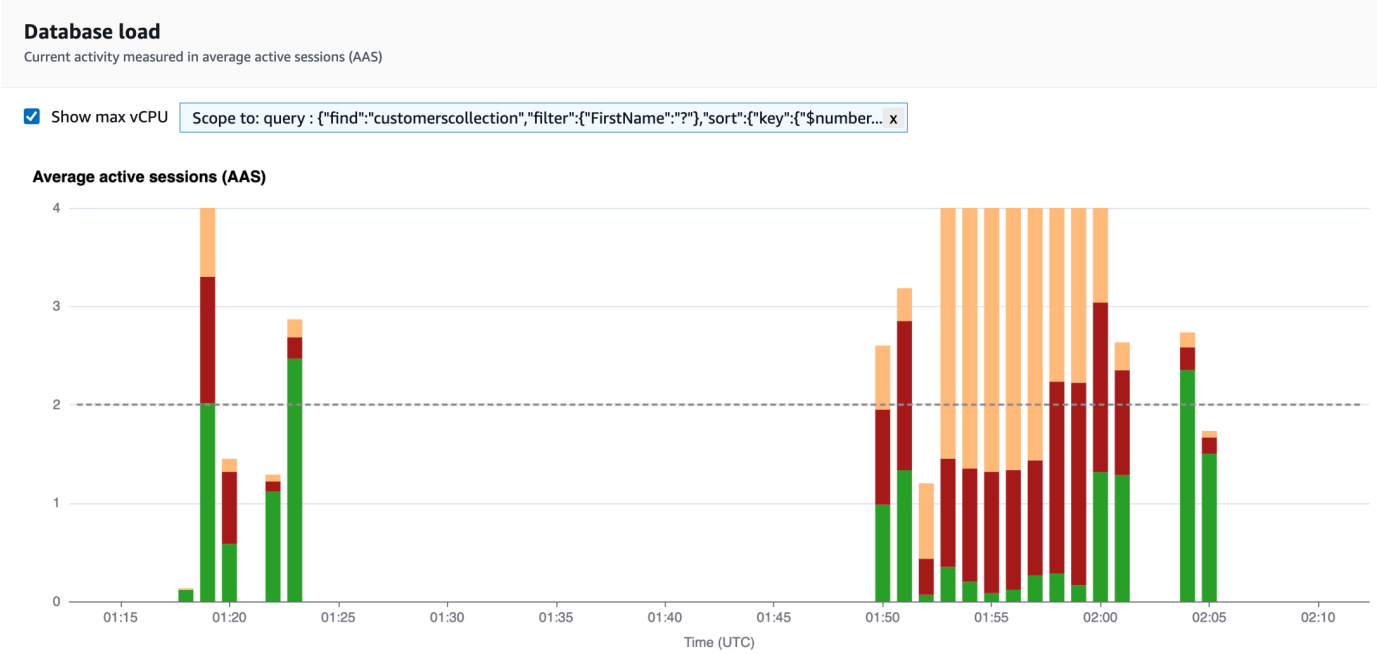
Für Amazon DocumentDB DocumentDB-Instances mit aktiviertem Performance Insights können Sie das Dashboard auch aufrufen, indem Sie in der Liste der Instances den Eintrag Sessions auswählen. Unter Aktuelle Aktivität zeigt das Element Sitzungen die Datenbanklast von durchschnittlichen, aktiven Sitzungen der letzten fünf Minuten an. Der Balken zeigt die Last grafisch an. Wenn die Leiste leer ist, befindet sich die Instance im Leerlauf. Wenn die Last ansteigt, wird der Balken blau ausgefüllt. Wenn die Last die Anzahl der virtuellen CPUs (vCPUs) in der Instance-Klasse überschreitet, färbt sich der Balken rot, was auf einen potenziellen Engpass hinweist.



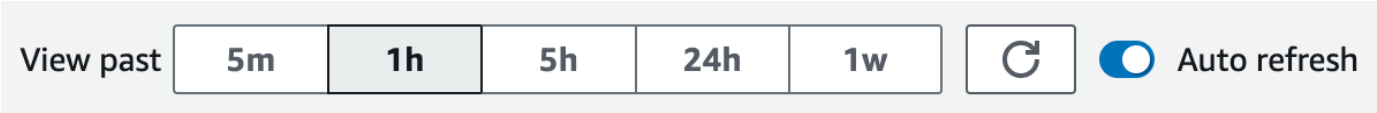
3. (Optional) Wählen Sie über eine der Schaltflächen rechts oben ein anderes Zeitintervall aus. Um das Intervall beispielsweise auf 1 Stunde zu ändern, wählen Sie 1h aus.



Im folgenden Screenshot beträgt das DB-Ladeintervall 1 Stunde.



4. Um Ihre Daten automatisch zu aktualisieren, aktivieren Sie Automatische Aktualisierung.



   Das Performance Insights-Dashboard wird automatisch mit neuen Daten aktualisiert. Die Aktualisierungsrate hängt von der Menge der angezeigten Daten ab:

   - „5 Minuten“ wird alle 5 Sekunden aktualisiert.
   - 1 Stunde, wird jede Minute aktualisiert.
   - 5 Stunden, wird jede Minute aktualisiert.
   - „24 Stunden“ wird alle 5 Minuten aktualisiert.
   - „1 Woche“ wird jede Stunden aktualisiert.

Analysieren der Datenbanklast anhand von Wartezuständen

Wenn das Diagramm zur Datenbankauslastung (DB-Last) einen Engpass anzeigt, können Sie herausfinden, woher die Last kommt. Betrachten Sie dazu die Tabelle mit den Hauptlastelementen unterhalb des Datenbanklast-Diagramms Wählen Sie ein bestimmtes Element aus, z. B. eine Abfrage oder eine Anwendung, um dieses Element genauer zu untersuchen und Details zu diesem Element anzuzeigen.

Die Datenbanklast, gruppiert nach Waits und Top-Abfragen, bietet in der Regel den besten Einblick in Leistungsprobleme. DB-Last gruppiert nach Wartezeiten zeigt an, ob Ressourcen- oder Parallelitätsengpässe in der Datenbank vorhanden sind. In diesem Fall wird auf der Registerkarte „Häufigste Abfragen“ der Tabelle mit den am häufigsten ausgelasteten Elementen angezeigt, welche Abfragen für diese Auslastung verantwortlich sind.

Ihr typischer Workflow für die Diagnose von Performance-Problemen ist folgendermaßen:

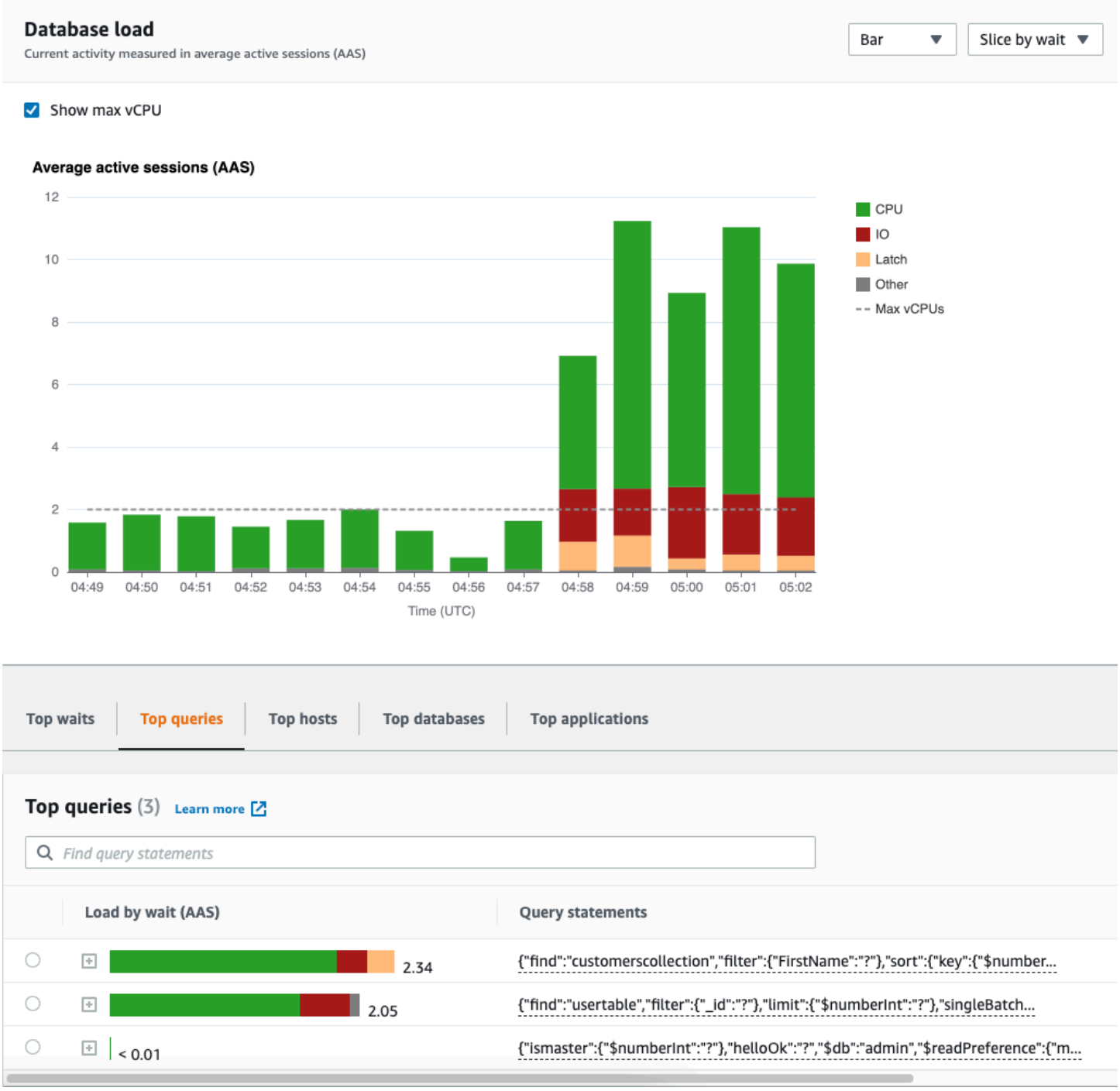
1. Überprüfen Sie das Diagramm der durchschnittlich aktiven Sitzungen auf irgendwelche Ereignisse, in denen die Datenbanklast die Max CPU-Linie übersteigt.
2. Wenn ja, schauen Sie sich das Diagramm der durchschnittlich aktiven Sitzungen an und identifizieren Sie, welcher Wartezustand oder welche Zustände primär dafür verantwortlich sind.
3. Identifizieren Sie die Digest-Abfragen, die die Last verursacht haben, indem Sie sehen, welche der Abfragen auf der Registerkarte „Häufigste Abfragen“ in der Tabelle mit den am häufigsten

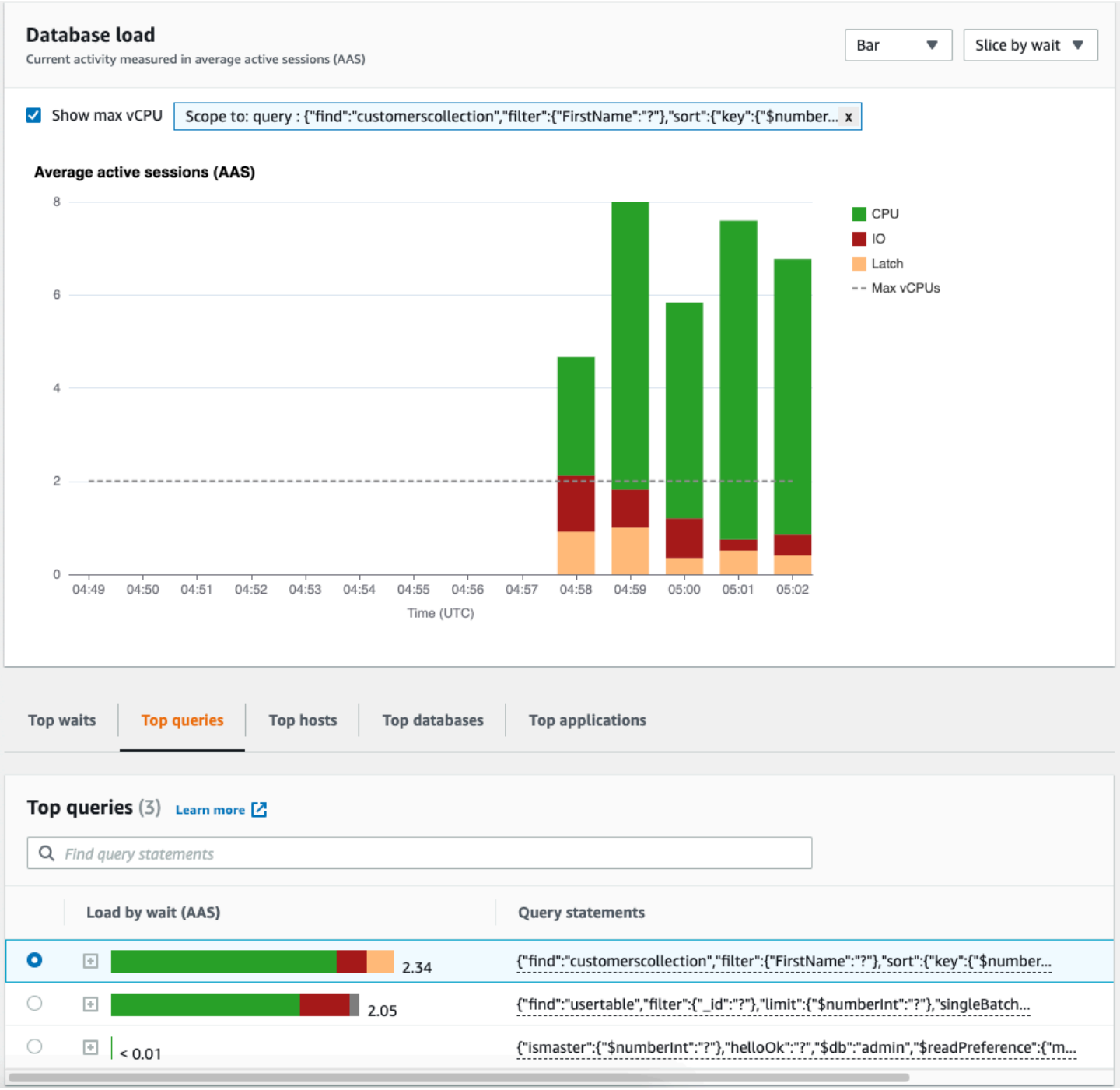
geladenen Elementen am meisten zu diesen Wartezuständen beitragen. Sie können diese anhand der Spalte Load by Wait (AAS) identifizieren.

4. Wählen Sie auf der Registerkarte „Häufig gestellte Abfragen“ eine dieser Digest-Abfragen aus, um sie zu erweitern und die untergeordneten Abfragen zu sehen, aus denen sie besteht.

Sie können auch sehen, welche Hosts oder Anwendungen die meiste Last verursachen, indem Sie jeweils Top-Hosts oder Top-Anwendungen auswählen. Anwendungsnamen werden in der Verbindungszeichenfolge zur Amazon DocumentDB DocumentDB-Instance angegeben. `Unknown`gibt an, dass das Anwendungsfeld nicht angegeben wurde.

Im folgenden Dashboard machen beispielsweise CPU-Wartezeiten den größten Teil der DB-Last aus. Wenn Sie unter Häufigste Abfragen die oberste Abfrage auswählen, wird das Diagramm zur Datenbankauslastung so ausgerichtet, dass der Schwerpunkt auf der höchsten Last liegt, die durch die ausgewählte Abfrage verursacht wird.


Database load
Current activity measured in average active sessions (AAS)
Bar
Slice by wait
Show max vCPU
Average active sessions (AAS)
12
10
8
6
4
2
0
CPU
IO
Latch
Other
Max vCPUs
04:49
04:50
04:51
04:52
04:53
04:54
04:55
04:56
04:57
04:58
04:59
05:00
05:01
05:02
Time (UTC)
Top waits
Top queries
Top hosts
Top databases
Top applications
Top queries (3)
Learn more
Find query statements
Load by wait (AAS)
Query statements
2.34
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"$number...
2.05
{"find":"usertable","filter":{"_id":"?"},"limit":{"$numberInt":"?"},"singleBatch...
< 0.01
{"ismaster":{"$numberInt":"?"},"helloOk":"?","$db":"admin","$readPreference":{"m...



Überblick über die Registerkarte „Häufigste Abfragen“

Standardmäßig werden auf der Registerkarte „Häufigste Abfragen“ die Abfragen angezeigt, die am meisten zur DB-Auslastung beitragen. Sie können den Abfragetext analysieren, um Ihre Abfragen zu optimieren.

Themen

- Zusammenfassungen von Abfragen
- Nach Waits laden (AAS)
- Detaillierte Abfrageinformationen anzeigen
- Zugriff auf den Abfragetext der Anweisung
- Abfragetext für Kontoauszüge anzeigen und herunterladen

Zusammenfassungen von Abfragen

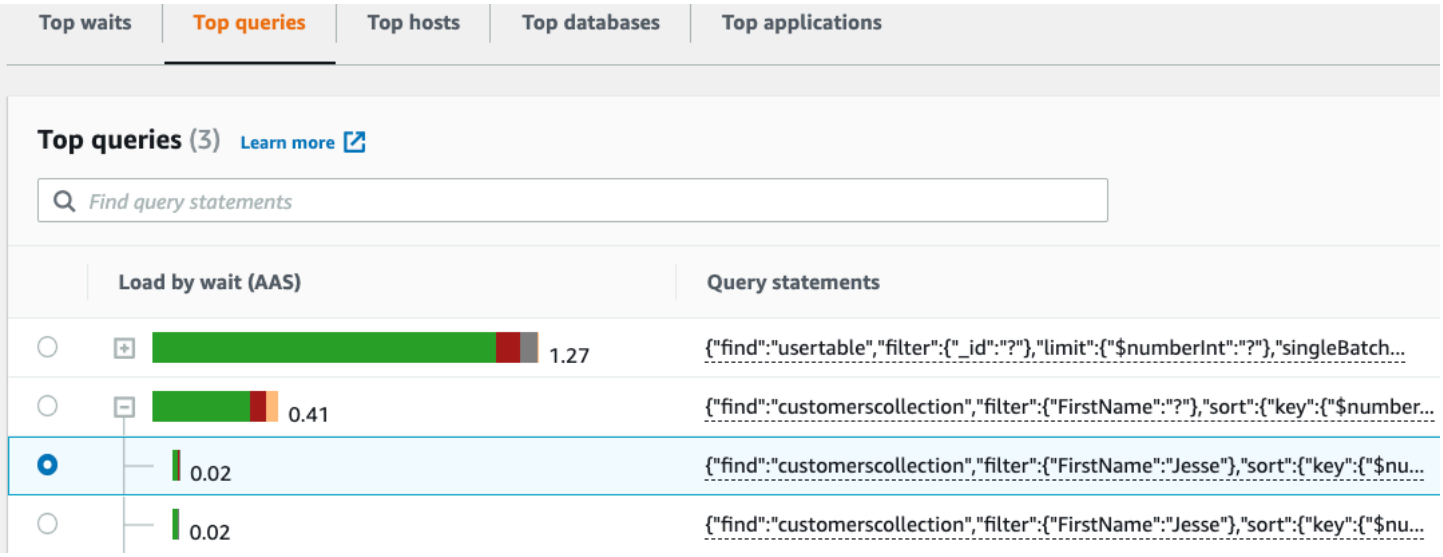
Ein Abfrage-Digest besteht aus mehreren tatsächlichen Abfragen, die sich strukturell ähneln, aber unterschiedliche Literalwerte haben können. Der Digest ersetzt fest codierte Werte durch ein Fragezeichen. Ein Abfrage-Digest könnte beispielsweise so aussehen:

```
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":
{"$numberInt":"?"}},"limit":{"$numberInt":"?"}}
```

Dieser Digest kann die folgenden untergeordneten Abfragen enthalten:

```
{"find":"customerscollection","filter":{"FirstName":"Karrie"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Met"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Rashin"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
```

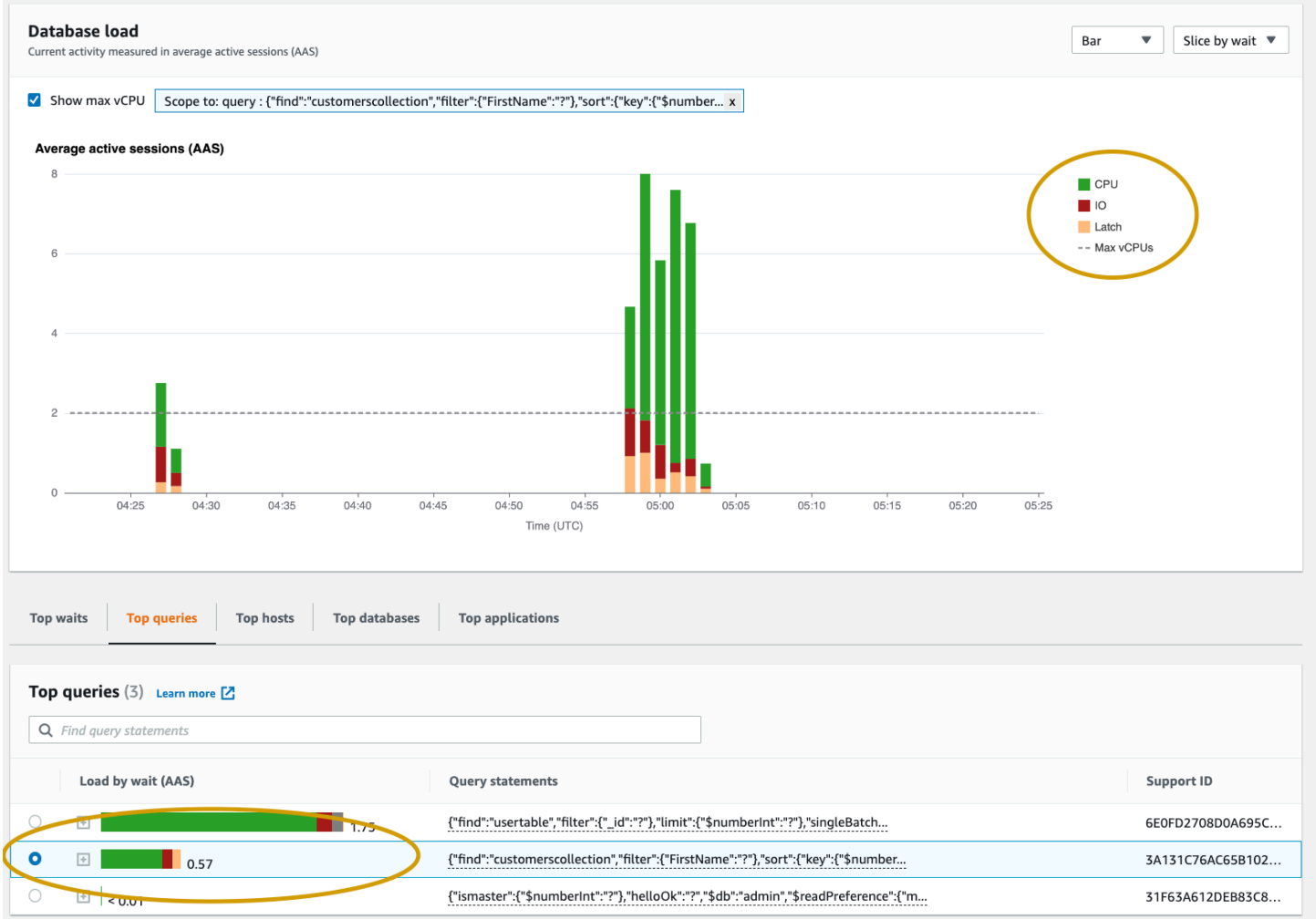
Um die wörtlichen Abfrageanweisungen in einem Digest zu sehen, wählen Sie die Abfrage und dann das Pluszeichen () aus. + Im folgenden Screenshot ist die ausgewählte Abfrage ein Digest.



> **Note**
>
> In einem Abfrage-Digest werden ähnliche Abfrageanweisungen gruppiert, sensible Informationen werden jedoch nicht geschwärzt.

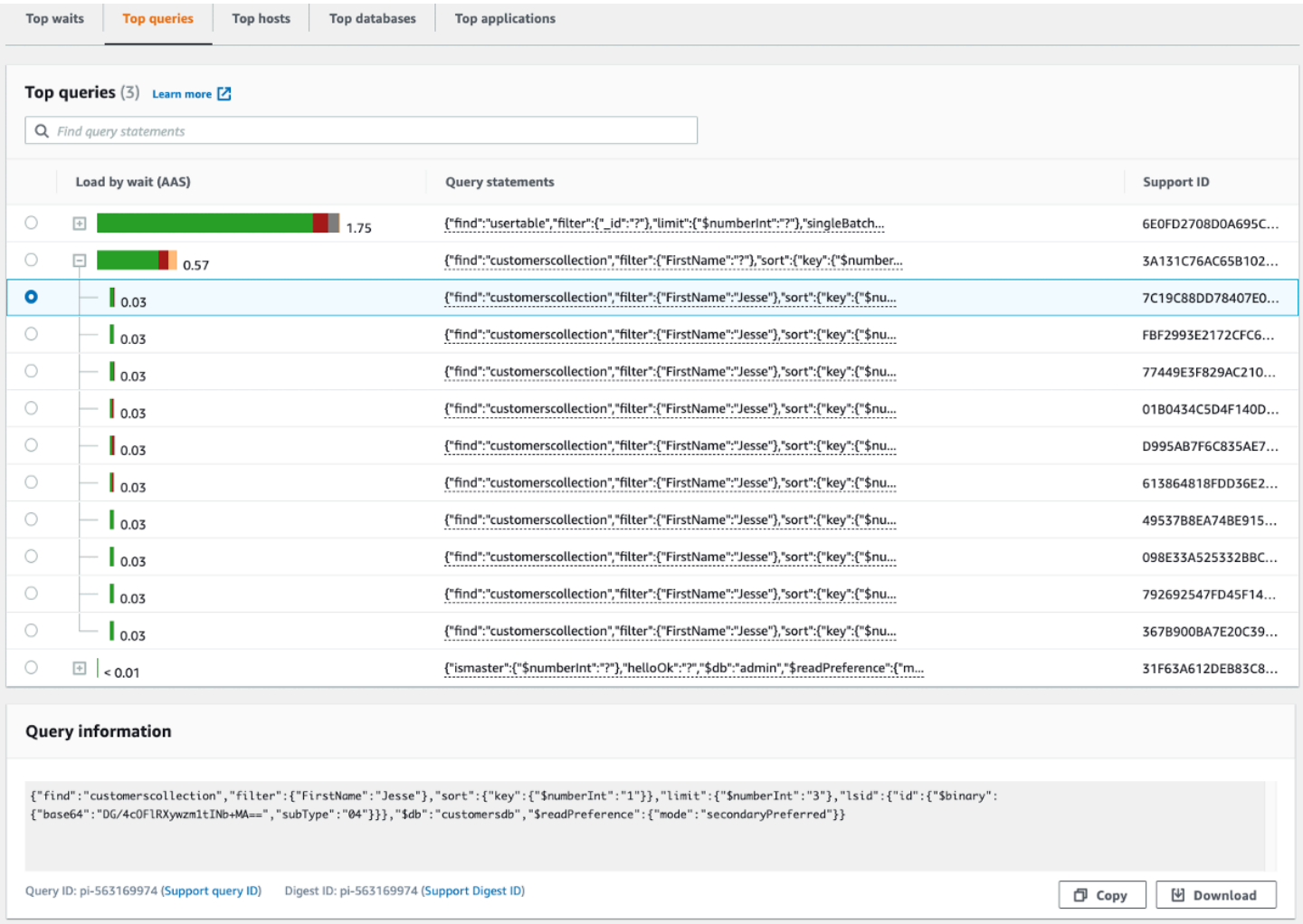
Nach Waits laden (AAS)

In Top-Abfragen zeigt die Spalte Load by Waits (AAS) den Prozentsatz der Datenbanklast, der jedem Top-Load-Element zugeordnet ist. Diese Spalte gibt die Last für dieses Element nach der Gruppierung wieder, die derzeit im DB-Lastdiagramm ausgewählt ist. Beispielsweise können Sie das DB-Last-Diagramm nach Wartezuständen gruppieren. In diesem Fall ist der Balken DB Load by Waits (DB-Last nach Wartezuständen) so groß, segmentiert und farbcodiert, dass angezeigt wird, zu wieviel Prozent diese Abfrage zum betreffenden Wartezustand beiträgt. Es zeigt zudem auf, welche Wartezustände sich auf die ausgewählte Abfrage auswirken.



Detaillierte Abfrageinformationen anzeigen

In der Abfragetabelle „Top“ können Sie eine Digest-Anweisung öffnen, um die zugehörigen Informationen anzuzeigen. Die Informationen werden im unteren Bereich angezeigt.



Die folgenden Typen von Bezeichnern (IDs) sind mit Abfrageanweisungen verknüpft:

1. Support-Abfrage-ID — Ein Hashwert der Abfrage-ID. Dieser Wert dient nur zum Verweisen auf eine Abfrage-ID, wenn Sie mit AWS Support arbeiten. AWS Der Support hat keinen Zugriff auf Ihre tatsächlichen Abfrage-IDs und Ihren Abfragetext.
2. Support-Digest-ID — Ein Hashwert der Digest-ID. Sie können eine Digest-ID über diesen Wert nur referenzieren, wenn Sie mit AWS Support arbeiten. AWS Der Support hat keinen Zugriff auf Ihre tatsächlichen Digest-IDs und den Abfragetext.

Zugriff auf den Abfragetext der Anweisung

Standardmäßig enthält jede Zeile in der Tabelle mit den häufigsten Abfragen 500 Byte Abfragetext für jede Abfrageanweisung. Wenn eine Digest-Anweisung 500 Byte überschreitet, können Sie mehr Text anzeigen, indem Sie die Anweisung im Performance Insights Insights-Dashboard öffnen. In

diesem Fall beträgt die maximale Länge der angezeigten Abfrage 1 KB. Wenn Sie eine vollständige Abfrageanweisung anzeigen, können Sie auch Herunterladen wählen.

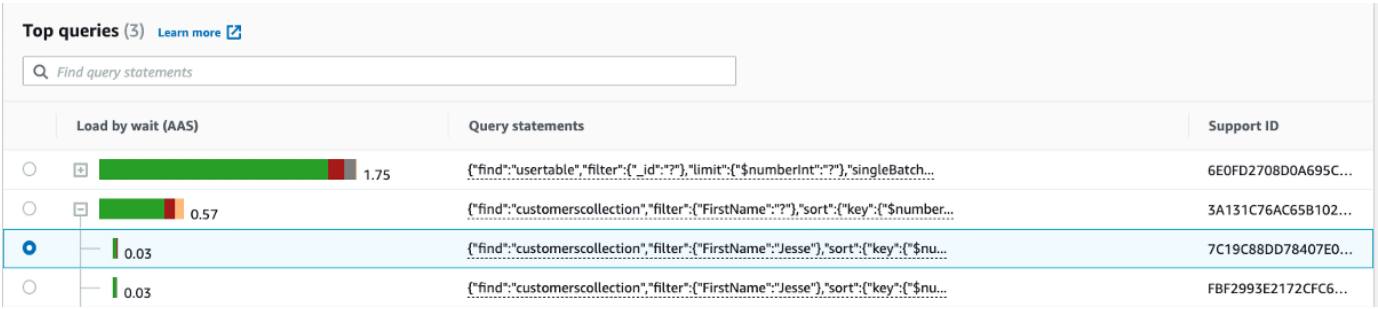
Abfragetext für Kontoauszüge anzeigen und herunterladen

Im Performance Insights Insights-Dashboard können Sie Abfragetext anzeigen oder herunterladen.

Um mehr Abfragetext im Performance Insights Insights-Dashboard anzuzeigen

1. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter: https://console.aws.amazon.com/docdb/
2. Wählen Sie im Navigationsbereich Performance-Insights aus.
3. Wählen Sie eine DB-Instance aus. Das Performance Insights-Dashboard wird für diese DB-Instance angezeigt.

   Abfrageanweisungen mit Text, der größer als 500 Byte ist, sehen wie in der folgenden Abbildung aus:



4. Sehen Sie sich den Abschnitt mit den Abfrageinformationen an, um mehr vom Abfragetext zu sehen.

Das Performance Insights Insights-Dashboard kann bis zu 1 KB für jede vollständige Abfrageanweisung anzeigen.

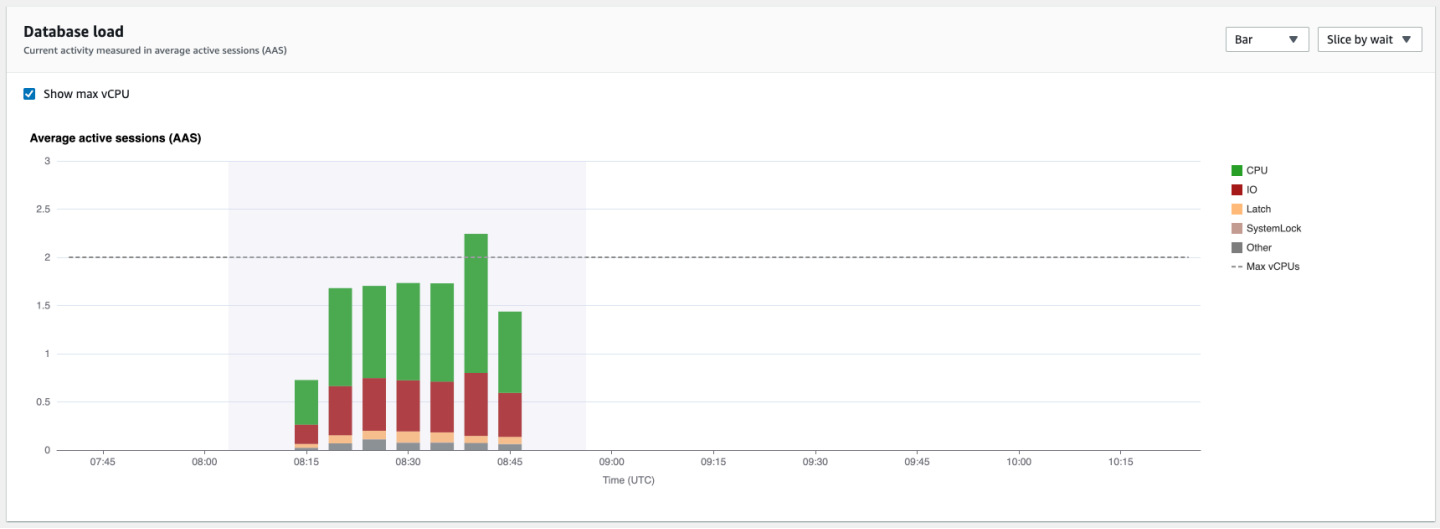
> **Note**
>
> Um die Abfrageanweisung zu kopieren oder herunterzuladen, deaktivieren Sie alle Popup-Blocker.

Das Diagramm zum Laden der Datenbank vergrößern

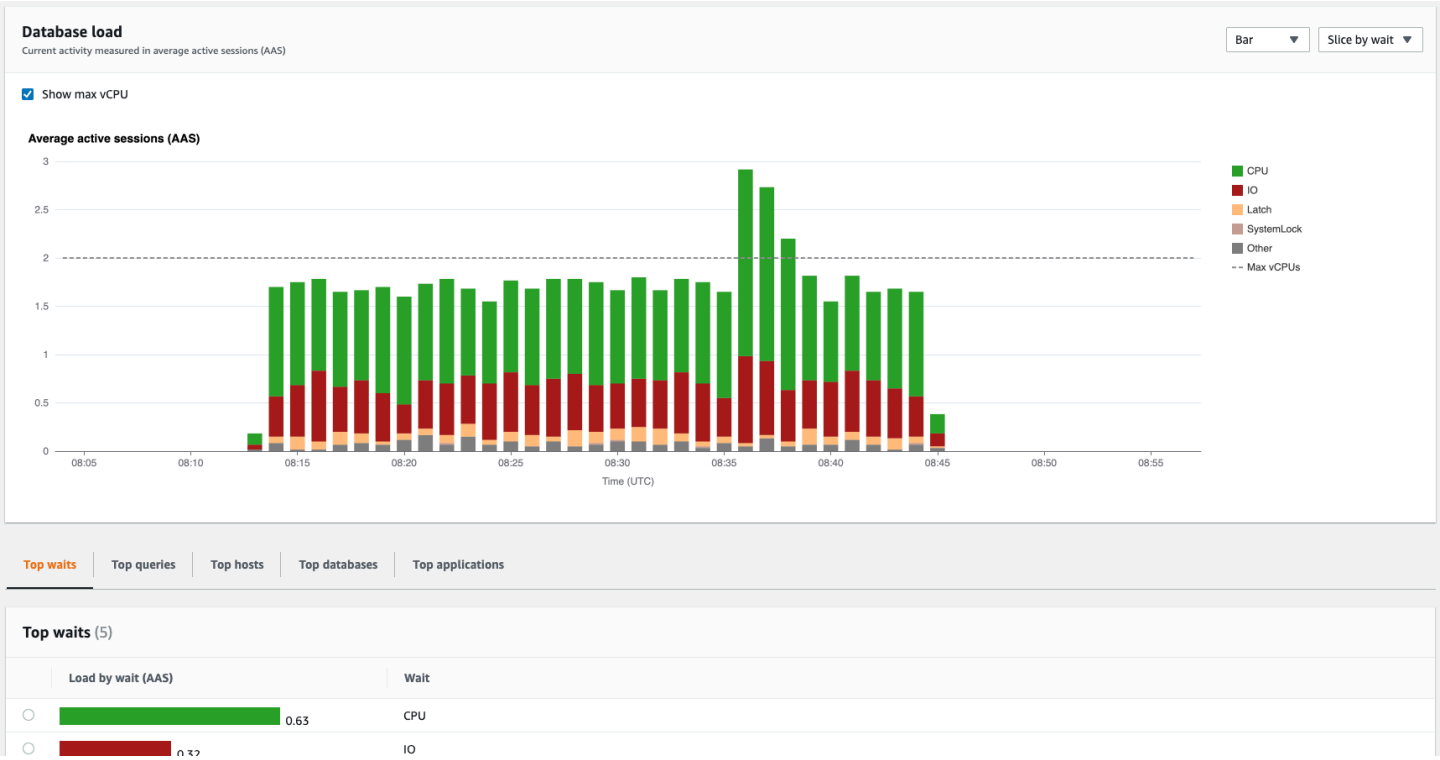
Sie können weitere Funktionen der Benutzeroberfläche von Performance Insights verwenden, um die Performance-Daten zu analysieren.

Click-and-Drag Zoom In

In der Benutzeroberfläche von Performance Insights können Sie einen kleinen Teil des Lastdiagramms auswählen und die Details vergrößern.



Um einen Teil des Lastdiagramms zu vergrößern, wählen Sie die Startzeit und ziehen Sie mit der Maus an das Ende des gewünschten Zeitraums. Dabei wird der markierte Bereich farblich hervorgehoben. Wenn Sie die Maustaste loslassen, vergrößert das Lastdiagramm den ausgewählten Bereich, und die Tabelle mit den wichtigsten Elementen wird neu berechnet.



Abrufen von Metriken mit der Performance Insights-API

Wenn Performance Insights aktiviert ist, bietet die API Einblicke in die Instance-Leistung Amazon CloudWatch Logs ist die maßgebliche Quelle für angebotene Monitoring-Metriken für AWS Services.

Performance Insights bietet eine domänenspezifische Ansicht der Datenbanklast, gemessen als durchschnittliche aktive Sitzungen (AAS). Diese Metrik erscheint API-Verbrauchern als zweidimensionaler Zeitreihendatensatz. Die Zeitdimension der Daten stellt die Datenbanklastdaten für jeden Zeitpunkt im abgefragten Zeitraum bereit. Für jeden Zeitpunkt wird die Gesamtlast bezogen auf die angeforderten Dimensionen zerlegt, z. B. `Query`, `Wait-state`, `Application` oder `Host`, gemessen zum betreffenden Zeitpunkt.

Amazon DocumentDB Performance Insights überwacht Ihre Amazon DocumentDB-DB-Instance, sodass Sie die Datenbankleistung analysieren und Fehler beheben können. Eine Möglichkeit zum Anzeigen von Performance Insights-Daten bietet die AWS Management Console. Performance Insights stellt außerdem eine öffentliche API bereit, sodass Sie Ihre eigenen Daten abfragen können. Sie können die API für Folgendes verwenden:

- Auslagern von Daten in eine Datenbank
- Hinzufügen von Performance Insights-Daten zu bestehenden Überwachungs-Dashboards
- Entwickeln von Überwachungstools

Um die Performance Insights-API zu verwenden, aktivieren Sie Performance Insights auf einer Ihrer Amazon DocumentDB DocumentDB-Instances. Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Aktivieren und Deaktivieren von Performance Insights. Weitere Informationen zur Performance Insights-API finden Sie in der Referenz zur Performance Insights-API.

Die Performance Insights-API bietet die folgenden Operationen.

Performance-Insights-Aktion	AWS CLI command	Beschreibung
DescribeDimensionKeys	aws pi describe-dimension-keys	Ruft die Schlüssel der Top N-Dimension für eine Metrik für einen bestimmten Zeitraum ab.
GetDimensionKeyDetails	aws pi get-dimension-key-details	Ruft die Attribute der angegebenen Dimensionsgruppe für eine DB-Instance oder Datenquelle ab. Wenn Sie beispielsweise eine Abfrage-ID angeben und die Dimensionsdetails verfügbar sind, wird der vollständige Text der mit dieser ID `db.query.statement` verknüpften Dimension `GetDimensionKeyDetails` abgerufen. Dieser Vorgang ist nützlich, weil er `GetResourceMetrics` das Abrufen von umfangreichem Text in Abfrageanweisungen `DescribeDimensionKeys` nicht unterstützt.
GetResourceMetadata	aws pi get-resource-metadata	Rufen Sie die Metadaten für verschiedene Funktionen ab. Die Metadaten könnten

Performance-Insights-Aktion	AWS CLI command	Beschreibung
		beispielsweise darauf hindeuten, dass eine Funktion für eine bestimmte DB-Instan ce ein- oder ausgeschaltet ist.
GetResourceMetrics	aws pi get-resource-metrics	Ruft Performance Insights-Metriken für eine Reihe von Datenquellen über einen Zeitraum ab. Sie können spezifische Dimension sgruppen und Dimensionen bereitstellen und Aggregati on und Filterkriterien für jede Gruppe bereitstellen.
ListAvailableResou rceDimensions	aws pi list-avai lable-resource-dim ensions	Rufen Sie die Dimensionen ab, die für jeden angegebenen Metriktyp für eine bestimmte Instance abgefragt werden können.
ListAvailableResou rceMetrics	aws pi list-avai lable-resource-met rics	Rufen Sie alle verfügbaren Metriken der angegeben en Metriktypen ab, die für eine bestimmte DB-Instance abgefragt werden können.

Themen

- AWS CLI für Performance Insights
- Abrufen von Zeitreihenmetriken
- AWS CLI-Beispiele für Performance-Insights

AWS CLI für Performance Insights

Sie können Performance Insights-Daten über die anzeige AWS CLI. Hilfe zu den AWS CLI-Befehlen für Performance Insights erhalten Sie durch Eingabe der folgenden Befehle an der Befehlszeile.

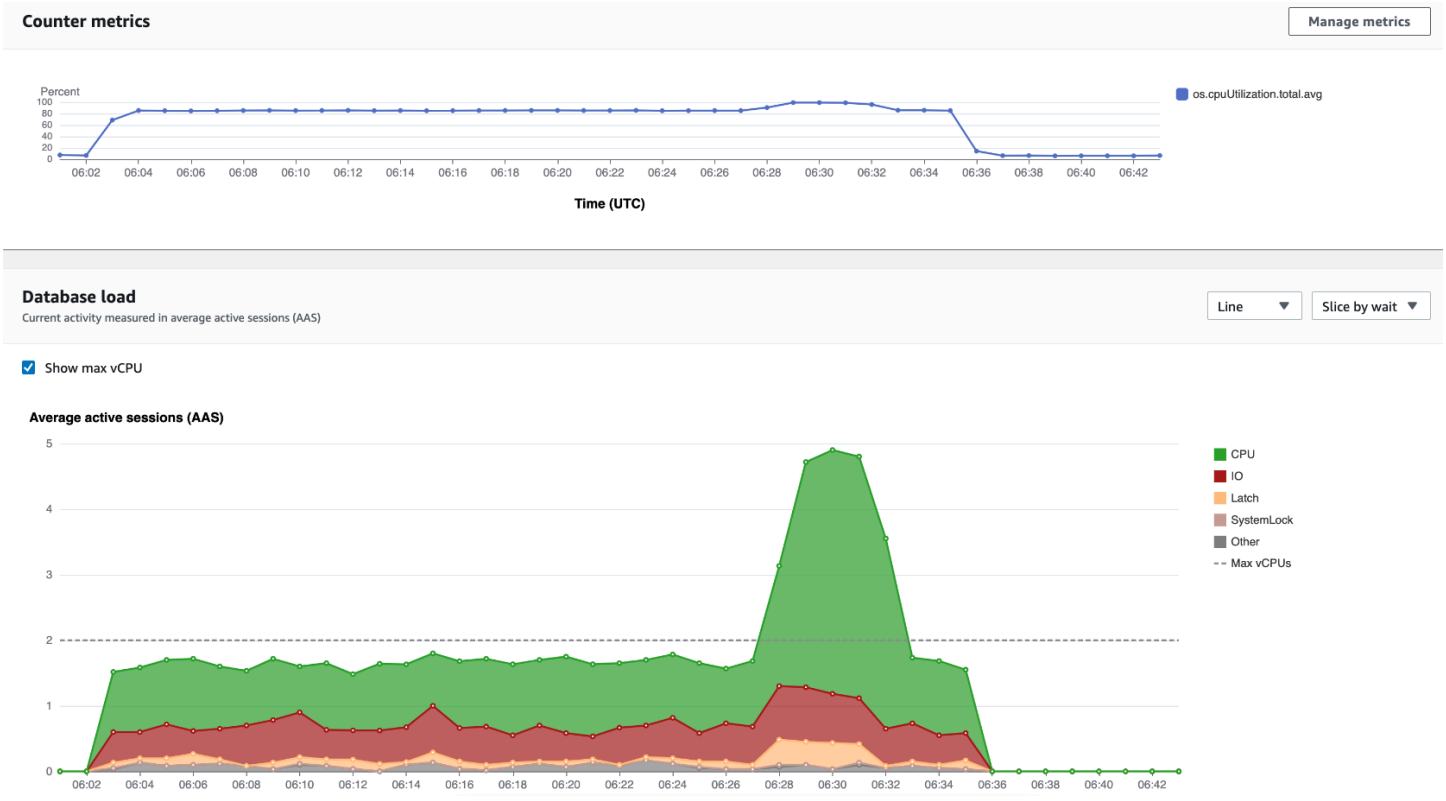
```
aws pi help
```

Wenn die AWS CLI nicht installiert ist, finden SieInformationen zur Installation unter Installieren der AWS-Befehlszeilenschnittstelle im AWS CLI-Benutzerhandbuch.

Abrufen von Zeitreihenmetriken

Mit der `GetResourceMetrics`-Operation werden ein oder mehrere Zeitreihenmetriken aus den Performance Insights-Daten abgerufen. Für `GetResourceMetrics` ist eine Metrik und ein Zeitraum erforderlich, damit eine Antwort mit einer Liste von Datenpunkten zurückgegeben wird.

Zum Beispiel dieAWS Management Console VerwendungGetResourceMetrics zum Ausfüllen des Diagramms „Zählermetriken“ und des Datenbank-Load-Diagramms, wie in der folgenden Abbildung dargestellt.



Alle von zurückgegebenen MetrikenGetResourceMetrics sind Standard-Zeitreihenmetriken, mit Ausnahme vondb.load. Diese Metrik wird im Diagramm Database Load (Datenbanklast)

angezeigt. Die`db.load` Metrik unterscheidet sich von den anderen Zeitreihenmetriken, da Sie sie in Unterkomponenten aufteilen können, die als Dimensionen bezeichnet werden. In der vorherigen Abbildung wird `db.load` unterteilt und nach Wartezuständen gruppiert, aus denen `db.load` besteht.

> **Note**
>
> `GetResourceMetrics` kann auch die `db.sampleload`-Metrik zurückgeben, aber die `db.load`-Metrik ist in den meisten Fällen angemessen.

Informationen zu den Zählermetriken, die von `GetResourceMetrics` zurückgegeben werden, finden Sie unter Performance Insights für Zählermetriken.

Die folgenden Berechnungen werden für die Metriken unterstützt:

- Durchschnitt – Der durchschnittliche Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.avg` an.
- Minimum – Der minimale Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.min` an.
- Maximum – Der maximale Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.max` an.
- Summe – Die Summe der Metrikwerte über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.sum` an.
- Beispielanzahl – Die Anzahl, wie oft die Metrik über einen bestimmten Zeitraum erfasst wurde. Fügen Sie dem Metriknamen `.sample_count` an.

Nehmen wir an, dass eine Metrik beispielsweise 300 Sekunden (5 Minuten) lang erfasst wird und dass die Metrik einmal pro Minute erfasst wird. Die Werte für jede Minute sind 1, 2, 3, 4 und 5. In diesem Fall werden die folgenden Berechnungen zurückgegeben:

- Durchschnitt – 3
- Minimum – 1
- Maximum – 5
- Summe – 15
- Beispielanzahl – 5

Weitere Informationen zur Verwendung des AWS CLI-Befehls `get-resource-metrics` finden Sie unter `get-resource-metrics`.

Geben Sie für die `--metric-queries`-Option eine oder mehrere Abfragen an, um die entsprechenden Ergebnisse zu erhalten. Jede Abfrage besteht aus einem obligatorischen `Metric`- sowie optionalen `GroupBy`- und `Filter`-Parametern. Es folgt ein Beispiel für eine Spezifikation der `--metric-queries`-Option.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

AWS CLI-Beispiele für Performance-Insights

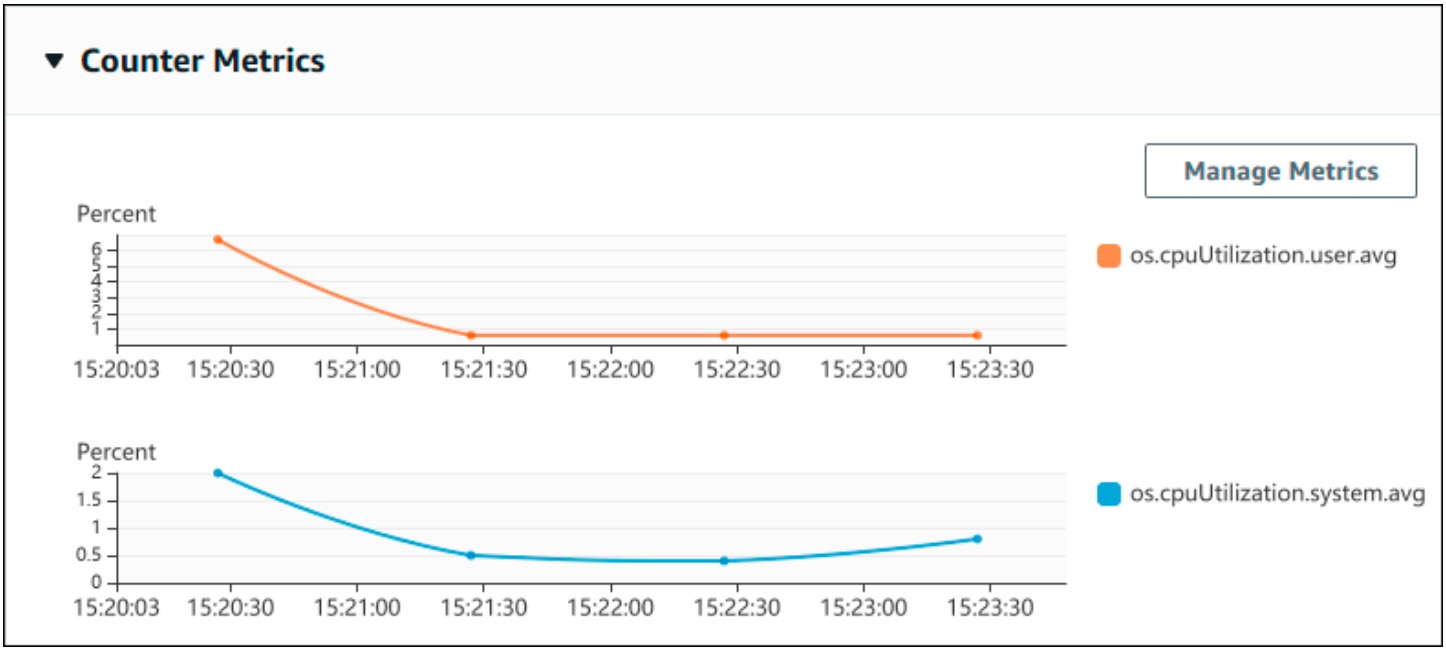
Die folgenden Beispiele zeigen, wie AWS CLI für Performance Insights verwendet wird.

Themen

- Abrufen von Zählermetriken
- Der DB-Lastdurchschnitt für die höchsten Wartezeiten wird abgerufen
- Der durchschnittliche DB-Ladestand für die oberste Abfrage wird abgerufen
- Abrufen des nach Query gefilterten DB-Lastdurchschnitts

Abrufen von Zählermetriken

Der folgende Screenshot zeigt zwei Zählermetriken-Diagramme in der AWS Management Console.



Das folgende Beispiel veranschaulicht, wie die Daten, die die AWS Management Console zum Erstellen der beiden Zählermetriken-Diagramme verwendet, gesammelt werden.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Sie können einen Befehl besser lesbar gestalten, indem Sie eine Datei für die Option `--metrics-query` angeben. Im folgenden Beispiel wird eine Datei namens query.json für die Option verwendet. Die Datei enthält Folgendes.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Führen Sie den folgenden Befehl aus, um die Datei zu verwenden.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

Das vorige Beispiel gibt die folgenden Werte für die Optionen an:

- `--service-type`— DOCDB für Amazon DocumentDB
- `--identifier` – Die Ressource-ID für die DB-Instance
- `--start-time` und `--end-time` – Die ISO 8601-Werte `DateTime` für den abzufragenden Zeitraum mit mehreren unterstützten Formaten

Der Abfragezeitraum beträgt eine Stunde:

- `--period-in-seconds` – `60` für eine Abfrage pro Minute
- `--metric-queries` – Ein Array mit zwei Abfragen, jeweils für nur eine Metrik.

  Der Metrikname verwendet Punkte, um die Metrik in eine sinnvolle Kategorie einzustufen, wobei das letzte Element eine Funktion ist. Im Beispiel lautet die Funktion `avg` für jede Abfrage. Wie bei Amazon CloudWatch sind die unterstützten Funktionen`min`, `max`, `total`, und`avg`.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": "2022-03-13T08:00:00+00:00",
    "AlignedEndTime": "2022-03-13T09:00:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {
            "Key": {
                "Metric": "os.cpuUtilization.user.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-03-13T08:01:00+00:00", //Minute1
                    "Value": 3.6
                },
                {
                    "Timestamp": "2022-03-13T08:02:00+00:00", //Minute2
                    "Value": 2.6
                },
                //.... 60 datapoints for the os.cpuUtilization.user.avg metric
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-03-13T08:01:00+00:00",
                    "Value": 92.7
                },
                {
                    "Timestamp": "2022-03-13T08:02:00+00:00",
                    "Value": 93.7
```

```
                },
                //.... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

Die Antwort enthält Werte für `Identifier`, `AlignedStartTime` und `AlignedEndTime`. Bei einem `--period-in-seconds`-Wert von `60` wurden Start- und Endzeiten auf die Minute ausgerichtet. Wenn der `--period-in-seconds`-Wert `3600` lautet, werden Start- und Endzeiten auf die Stunde ausgerichtet.

Die `MetricList` in der Antwort enthält eine Reihe von Einträgen, und zwar jeweils mit einem `Key`- und einem `DataPoints`-Eintrag. Jeder `DataPoint` verfügt über einen `Timestamp` und einen `Value`. Jede `Datapoints`-Liste enthält 60 Datenpunkte, da die Abfragen eine Stunde lang jede Minute Daten abfragen, und zwar mit den Werten `Timestamp1/Minute1`, `Timestamp2/Minute2` usw. bis `Timestamp60/Minute60`.

Da sich die Abfrage auf zwei verschiedene Zählermetriken bezieht, enthält die -Antwort zwei Element `MetricList`.

Der DB-Lastdurchschnitt für die höchsten Wartezeiten wird abgerufen

Das folgende Beispiel entspricht der Abfrage, die die AWS Management Console zum Erstellen eines Stapelflächendiagramms verwendet. In diesem Beispiel wird der Wert `db.load.avg` für die letzte Stunde abgerufen, wobei die Last nach den sieben höchsten Wartezuständen aufgeteilt wird. Der Befehl ist mit dem Befehl unter identisc Abrufen von Zählermetriken. Die Datei query.json enthält hingegen Folgendes.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 7 }
    }
]
```

Führen Sie den folgenden Befehl aus.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
```

```
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

Das Beispiel gibt die Metrik für `db.load.avg` und a `GroupBy` der sieben wichtigsten Wartezustände an. Einzelheiten zu gültigen Werten für dieses Beispiel finden Sie DimensionGroupin der Performance Insights API-Referenz.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": "2022-04-04T06:00:00+00:00",
    "AlignedEndTime": "2022-04-04T06:15:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {//A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
                    "Value": 0.0
                },
                {
                    "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
```

```
                    "Value": 0.0
                },
                //... 60 datapoints for the total db.load.avg key
                ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_state.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
                    "Value": 0.0
                },
                {
                    "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
                    "Value": 0.0
                },
                //... 60 datapoints for the CPU key
            ]
        },//... In total we have 3 key/datapoints entries, 1) total, 2-3) Top Wait
 States
    ] //end of MetricList
} //end of response
```

In dieser Antwort gibt es drei Einträge in der`MetricList`. Es gibt einen Eintrag für die Gesamtzahl `db.load.avg` und jeweils drei Einträge für die `db.load.avg` Aufteilung nach einem der drei höchsten Wartezustände. Da es eine Gruppierungsdimension gab (im Gegensatz zum ersten Beispiel), muss es für jede Gruppierung der Metrik einen Schlüssel geben. Für jede Metrik kann nicht nur ein Schlüssel vorhanden sein, wie im Anwendungsfall der Basiszählermetrik.

Der durchschnittliche DB-Ladestand für die oberste Abfrage wird abgerufen

Das folgende Beispiel `db.wait_state` gruppiert nach den 10 wichtigsten Abfrageanweisungen. Es gibt zwei verschiedene Gruppen für Abfrageanweisungen:

- `db.query`— Die vollständige Abfrageanweisung, wie `{"find":"customers","filter": {"FirstName":"Jesse"},"sort":{"key":{"$numberInt":"1"}}}`

- `db.query_tokenized`— Die tokenisierte Abfrageanweisung, wie `{"find":"customers","filter":{"FirstName":"?"},"sort":{"key": {"$numberInt":"?"}},"limit":{"$numberInt":"?"}}`

Bei der Analyse der Datenbankleistung kann es nützlich sein, Abfrageanweisungen, die sich nur durch ihre Parameter unterscheiden, als ein Logikelement zu betrachten. In diesem Fall können Sie `db.query_tokenized` beim Abfragen verwenden. Vor allem, wenn Sie daran interessiert sind`explain()`, ist es manchmal sinnvoller, vollständige Abfrageanweisungen mit Parametern zu untersuchen. Es besteht eine Beziehung zwischen tokenisierten und vollständigen Abfragen, wobei mehrere vollständige Abfragen (untergeordnete Abfragen) unter derselben tokenisierten Abfrage (übergeordnete Abfrage) gruppiert sind.

Der Befehl in diesem Beispiel ähnelt dem Befehl unter Der DB-Lastdurchschnitt für die höchsten Wartezeiten wird abgerufen. Die Datei query.json enthält hingegen Folgendes.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.query_tokenized", "Limit": 10 }
    }
]
```

Im folgenden Beispiel wird verwende `db.query_tokenized`.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
```

```
    --end-time   2022-03-13T9:00:00Z  ^
    --period-in-seconds 3600 ^
    --metric-queries file://query.json
```

In diesem Beispiel werden Abfragen über eine Stunde mit einer Minute abgefragt. period-in-seconds

Das Beispiel gibt die Metrik für `db.load.avg` und einen `GroupBy` der sieben höchsten Wartezustände an. Einzelheiten zu gültigen Werten für dieses Beispiel finden Sie DimensionGroupin der Performance Insights API-Referenz.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": "2022-04-04T06:00:00+00:00",
    "AlignedEndTime": "2022-04-04T06:15:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {//A list of key/datapoints
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //... 60 datapoints for the total db.load.avg key
                ]
        },
               {
            "Key": {//Next key are the top tokenized queries
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.query_tokenized.db_id": "pi-1064184600",
                    "db.query_tokenized.id": "77DE8364594EXAMPLE",
                    "db.query_tokenized.statement": "{\"find\":\"customers\",\"filter
\":{\"FirstName\":\"?\"},\"sort\":{\"key\":{\"$numberInt\":\"?\"}},\"limit\"
:{\"$numberInt\":\"?\"},\"$db\":\"myDB\",\"$readPreference\":{\"mode\":\"primary\"}}"
                }
            },
            "DataPoints": [
            //... 60 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized queries, 1 total key
    ] //End of MetricList
} //End of response
```

Diese Antwort enthält 11 Einträge in der Abfrage `MetricList` (insgesamt 1, 10 am häufigsten tokenisierte Abfragen), wobei jeder Eintrag 24 Einträge pro Stunde enthält. `DataPoints`

Bei tokenisierten Abfragen gibt es drei Einträge in jeder Dimensionsliste:

- `db.query_tokenized.statement`— Die tokenisierte Abfrageanweisung.
- `db.query_tokenized.db_id` — Die synthetische ID, die Performance Insights für Sie generiert. In diesem Beispiel wird die synthetische ID `pi-1064184600` zurückgegeben.
- `db.query_tokenized.id` – Die ID der Abfrage innerhalb von Performance-Insights.

  In der AWS Management Console wird diese ID als Support-ID bezeichnet. Es wird so genannt, weil die ID Daten sind, die der AWS-Support untersuchen kann, um Ihnen bei der Behebung eines Problems mit Ihrer Datenbank zu helfen. AWS nimmt die Sicherheit und den Datenschutz Ihrer Daten sehr ernst und fast alle Daten werden mit Ihrem AWS KMS-Kundenstamm (CMK) verschlüsselt gespeichert. Daher sind diese Daten für keinen Benutzer innerhalb von AWS einsehbar. Im vorherigen Beispiel wird sowohl `tokenized.statement` als auch `tokenized.db_id` verschlüsselt gespeichert. Bei einem Problem mit Ihrer Datenbank kann der AWS Support Sie anhand der Support-ID unterstützen.

Beim Abfragen empfiehlt es sich ggf., eine `Group` in `GroupBy` anzugeben. Für eine präzisere Kontrolle der Daten, die zurückgegeben werden, sollten Sie allerdings die Dimensionsliste angeben. Wenn z. B. lediglich eine `db.query_tokenized.statement` erforderlich ist, kann der query.json-Datei ein `Dimensions`-Attribut hinzugefügt werden.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.query_tokenized",
            "Dimensions":["db.query_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Abrufen des nach Query gefilterten DB-Lastdurchschnitts

Die entsprechende API-Abfrage in diesem Beispiel ähnelt dem Befehl unter Der durchschnittliche DB-Ladestand für die oberste Abfrage wird abgerufen. Die Datei query.json enthält hingegen Folgendes.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 5  },
        "Filter": { "db.query_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
]
```

In dieser Antwort werden alle Werte entsprechend dem Beitrag der tokenisierten Abfrage AKIAIOSFODNN7EXAMPLE gefiltert, die in der Datei query.json angegeben ist. Die Schlüssel haben möglicherweise auch eine andere Reihenfolge als eine Abfrage ohne Filter, da sich die gefilterte Abfrage auf die fünf Wartezustände mit der höchsten Wartezeit ausgewirkt hat.

CloudWatch Amazon-Metriken für Performance Insights

Performance Insights veröffentlicht automatisch Metriken auf Amazon CloudWatch. Dieselben Daten können von Performance Insights abgefragt werden, aber wenn die Metriken vorhanden sind, ist es einfach, Alarme hinzuzufügen CloudWatch . CloudWatch Die Metriken können auch leicht zu vorhandenen CloudWatch-Dashboards hinzugefügt werden.

Metrik	Beschreibung
DBLoad	Die Anzahl der aktiven Sitzungen für Amazon DocumentDB. In der Regel sind Sie an den Daten für die durchschnittliche Anzahl der aktiven Sitzungen interessiert. Diese Daten werden in Performance Insights als abgefrag `db.load.avg` .
DBLoadCPU	Die Anzahl der aktiven Sitzungen, bei denen der Wartestatustyp CPU ist. In Performan ce Insights werden diese Daten abgefragt als`db.load.avg` , gefiltert nach dem Wartestatustyp. `CPU`
DB-CPU LoadNon	Die Anzahl der aktiven Sitzungen, bei denen der Wartestatustyp nicht CPU ist.

> **Note**
>
> Diese Metriken werden CloudWatch nur veröffentlicht, wenn die DB-Instance ausgelastet ist.

Sie können diese Metriken mithilfe der CloudWatch KonsoleAWS CLI, der oder der CloudWatch API untersuchen.

Sie können beispielsweise die Statistiken für die `DBLoad` Metrik abrufen, indem Sie den get-metric-statisticsBefehl ausführen.

```
aws cloudwatch get-metric-statistics \
    --region ap-south-1 \
    --namespace AWS/DocDB \
    --metric-name DBLoad  \
    --period 360 \
    --statistics Average \
    --start-time 2022-03-14T8:00:00Z \
    --end-time 2022-03-14T9:00:00Z \
    --dimensions Name=DBInstanceIdentifier,Value=documentdbinstance
```

Dieses Beispiel generiert eine Ausgabe wie die folgende.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-03-14T08:42:00Z",
            "Average": 1.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:24:00Z",
            "Average": 2.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:54:00Z",
            "Average": 6.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:36:00Z",
```

```
            "Average": 5.7,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:06:00Z",
            "Average": 4.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:00:00Z",
            "Average": 5.2,
            "Unit": "None"
        }
    ],
    "Label": "DBLoad"
}
```

Sie können die mathematische `DB_PERF_INSIGHTS` Metrikfunktion in der CloudWatch Konsole verwenden, um Zählermetriken von Amazon DocumentDB Performance Insights abzufragen. Die `DB_PERF_INSIGHTS` Funktion beinhaltet auch die `DBLoad` Metrik in Intervallen unter einer Minute. Sie können CloudWatch Alarme für diese Messwerte einrichten. Weitere Informationen zum Erstellen eines Alarms finden Sie unter Erstellen eines Alarms zu Performance Insights-Zählermetriken aus einer AWS-Datenbank.

Weitere Informationen zu CloudWatch finden Sie unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch.

Performance Insights für Zählermetriken

Zählermetriken sind Betriebssystemmetriken im Performance Insights Insights-Dashboard. Um Leistungsprobleme zu identifizieren und zu analysieren, können Sie Zählermetriken mit der DB-Last korrelieren.

Performance Insights-Betriebssystemzähler

Die folgenden Betriebssystemindikatoren sind mit DocumentDB Performance Insights verfügbar.

Zähler	Typ	Metrik
Aktiv	memory	os.memory.active

Zähler	Typ	Metrik
buffers	memory	os.memory.buffers
cached	memory	os.memory.cached
dirty	memory	os.memory.dirty
free	memory	os.memory.free
inactive	memory	os.memory.inactive
mapped	memory	os.memory.mapped
pageTables	memory	os.memory.pageTables
slab	memory	os.memory.slab
total	memory	os.memory.total
writeback	memory	os.memory.writeback
idle	cpuUtilization	os.cpuUtilization.idle
system	cpuUtilization	os.cpuUtilization.system
total	cpuUtilization	os.cpuUtilization.total
user	cpuUtilization	os.cpuUtilization.user
wait	cpuUtilization	os.cpuUtilization.wait
one	loadAverageMinute	OS. loadAverageMinute. eins
fifteen	loadAverageMinute	os. loadAverageMinute. fünfzehn
fünf	loadAverageMinute	os. loadAverageMinute. fünf
cached	swap	os.swap.cached
free	swap	os.swap.free

Zähler	Typ	Metrik
in	swap	os.swap.in
out	swap	os.swap.out
total	swap	os.swap.total
rx	network	os.network.rx
tx	network	os.network.tx
numVCPUs	general	os.general.numVCPUs

Entwickeln mit Amazon DocumentDB

Diese Abschnitte behandeln die Entwicklung mit Amazon DocumentDB (mit MongoDB-Kompatibilität).

Themen

Programmgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB

Dieser Abschnitt enthält Codebeispiele, die zeigen, wie Sie mithilfe verschiedener Sprachen eine Verbindung zu Amazon DocumentDB (mit MongoDB-Kompatibilität) herstellen können. Die Beispiele sind in zwei Abschnitte unterteilt: Verbindung mit einem Cluster mit und ohne aktiviertem TLS (Transport Layer Security). Standardmäßig ist TLS auf Amazon DocumentDB-Clustern aktiviert. Sie können TLS jedoch bei Bedarf deaktivieren. Weitere Informationen finden Sie unter Datenverschlüsselung während der Übertragung.

Wenn Sie versuchen, von außerhalb der VPC, in der sich Ihr Cluster befindet, eine Verbindung zu Ihrer Amazon DocumentDB herzustellen, lesen Sie bitte Verbindung zu einem Amazon DocumentDB-Cluster von außerhalb einer Amazon VPC herstellen.

Bevor Sie sich mit Ihrem Cluster verbinden, müssen Sie wissen, ob TLS auf dem Cluster aktiviert ist. Der nächste Abschnitt zeigt Ihnen, wie Sie den Wert des Parameters `tls` Ihres Clusters mit der AWS

Management Console oder der AWS CLI bestimmen können. Danach können Sie fortfahren, indem Sie das entsprechende Codebeispiel anwenden.

Themen

Bestimmen des Wertes des `tls`-Parameters

Zu bestimmen, ob Ihr Cluster TLS aktiviert hat, ist ein zweistufiger Prozess. Diesen können Sie mit der AWS Management Console oder AWS CLI durchführen.

1. Bestimmen Sie, welche Parametergruppe Ihren Cluster steuert.

   Using the AWS Management Console

   1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
   2. Wählen Sie im linken Navigationsbereich die Option Cluster aus.
   3. Wählen Sie in der Liste der Cluster den Namen des Clusters aus.
   4. Auf der anschließend angezeigten Seite werden die Details des von Ihnen ausgewählten Clusters angezeigt. Scrollen Sie nach unten zu Cluster details (Clusterdetails). Sie finden den Namen der Parametergruppe unten in diesem Abschnitt unterhalb von Cluster parameter group (Cluster-Parametergruppe).

   Using the AWS CLI

   Über folgenden AWS CLI-Code wird ermittelt, welcher Parameter Ihren Cluster steuert. Ersetzen Sie `sample-cluster` durch den Namen Ihres Clusters.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
   ```

   Die Ausgabe dieser Operation sieht in etwa wie folgt aus:

```
[
       [
           "sample-cluster",
           "sample-parameter-group"
       ]
]
```

2. Bestimmen Sie den Wert des Parameters **`tls`** in der Parametergruppe Ihres Clusters.

   Using the AWS Management Console

   1. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
   2. Wählen Sie im Fenster Cluster parameter groups (Cluster-Parametergruppen) Ihre Cluster-Parametergruppe aus.
   3. Auf der Ergebnisseite werden die Parameter der Cluster-Parametergruppe angezeigt. Hier können Sie den Wert des `tls`-Parameters sehen. Informationen zum Ändern dieses Parameters finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen ändern.

   Using the AWS CLI

   Sie können den Befehl `describe-db-cluster-parameters` AWS CLI verwenden, um sich die Details der Parameter in Ihrer Cluster-Parametergruppe anzeigen zu lassen.

   - **`--describe-db-cluster-parameters`**— Um alle Parameter innerhalb einer Parametergruppe und ihre Werte aufzulisten.
     - **`--db-cluster-parameter-group name`** – Erforderlich. Der Name Ihrer Cluster-Parametergruppe.

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Die Ausgabe dieser Operation sieht in etwa wie folgt aus:

   ```
   {
           "Parameters": [
               {
                   "ParameterName": "profiler_threshold_ms",
   ```

```
                "ParameterValue": "100",
                "Description": "Operations longer than profiler_threshold_ms
 will be logged",
                "Source": "system",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "50-2147483646",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            },
            {
                "ParameterName": "tls",
                "ParameterValue": "disabled",
                "Description": "Config to enable/disable TLS",
                "Source": "user",
                "ApplyType": "static",
                "DataType": "string",
                "AllowedValues": "disabled,enabled,fips-140-3",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            }
        ]
}
```

> **Note**
>
> Amazon DocumentDB unterstützt FIPS 140-3-Endpunkte, beginnend mit Amazon DocumentDB 5.0-Clustern (Engine-Version 3.0.3727) in diesen Regionen: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west

Nachdem Sie den Wert des `tls`-Parameters bestimmt haben, fahren Sie mit dem Herstellen einer Verbindung zu Ihrem Cluster fort, indem Sie eines der Codebeispiele in den folgenden Abschnitten verwenden.

- Verbindung bei aktiviertem TLS herstellen
- Verbinden bei deaktiviertem TLS

Verbindung bei aktiviertem TLS herstellen

Um ein Codebeispiel für die programmgesteuerte Verbindung zu einem TLS-fähigen Amazon DocumentDB-Cluster anzuzeigen, wählen Sie die entsprechende Registerkarte für die Sprache, die Sie verwenden möchten.

Um Daten während der Übertragung zu verschlüsseln, laden Sie den öffentlichen Schlüssel für Amazon DocumentDB `global-bundle.pem` mit dem folgenden Vorgang herunter.

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Wenn sich Ihre Anwendung auf Microsoft Windows befindet und eine PKCS7-Datei benötigt, können Sie das PKCS7-Zertifikat-Bundle herunterladen. Dieses Paket enthält sowohl das Zwischen- als auch das Stammzertifikat unter https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b.

Python

Der folgende Code zeigt, wie Sie mit Python eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
import pymongo
import sys

##Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
 and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
```

```
print(x)

##Close the connection
client.close()
```

Node.js

Der folgende Code zeigt, wie Sie mit Node.js eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
var MongoClient = require('mongodb').MongoClient

//Create a MongoDB client, open a connection to DocDB; as a replica set,
//  and specify the read preference as secondary preferred

var client = MongoClient.connect(
'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
tls=true&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  tlsCAFile: `global-bundle.pem` //Specify the DocDB; cert
},
function(err, client) {
    if(err)
        throw err;

    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'DocDB;'}, function(err, result){
        //Print the result to the screen
        console.log(result);

        //Close the connection
        client.close()
      });
   });
```

```
});
```

PHP

Der folgende Code zeigt, wie Sie mit PHP eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

$TLS_DIR = "/home/ubuntu/global-bundle.pem";

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoDB\Client("mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?retryWrites=false", ["tls" =>
 "true", "tlsCAFile" => $TLS_DIR ]);

//Specify the database and collection to be used
$col = $client->sampledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB'] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

Der folgende Code zeigt, wie Sie mit Go eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

> **Note**
>
> Ab Version 1.2.1 verwendet der MongoDB Go Driver nur das erste CA-Serverzertifikat, das in `sslcertificateauthorityfile` gefunden wurde. Der folgende Beispielcode behebt diese Einschränkung, indem alle in `sslcertificateauthorityfile`

gefundenen Serverzertifikate manuell an eine benutzerdefinierte TLS-Konfiguration angehängt werden, die während der Client-Erstellung verwendet wird.

```
package main

import (
 "context"
 "fmt"
 "log"
 "time"

 "go.mongodb.org/mongo-driver/bson"
 "go.mongodb.org/mongo-driver/mongo"
 "go.mongodb.org/mongo-driver/mongo/options"

 "io/ioutil"
 "crypto/tls"
 "crypto/x509"
 "errors"
)

const (
 // Path to the AWS CA file
 caFilePath = "global-bundle.pem"

 // Timeout operations after N seconds
 connectTimeout  = 5
 queryTimeout    = 30
 username        = "<sample-user>"
 password        = "<password>"
 clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

 // Which instances to read from
 readPreference = "secondaryPreferred"

 connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
tls=true&replicaSet=rs0&readpreference=%s"
)

func main() {
```

```
connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
clusterEndpoint, readPreference)

tlsConfig, err := getCustomTLSConfig(caFilePath)
if err != nil {
 log.Fatalf("Failed getting TLS configuration: %v", err)
}

client, err :=
mongo.NewClient(options.Client().ApplyURI(connectionURI).SetTLSConfig(tlsConfig))
if err != nil {
 log.Fatalf("Failed to create client: %v", err)
}

ctx, cancel := context.WithTimeout(context.Background(),
connectTimeout*time.Second)
defer cancel()

err = client.Connect(ctx)
if err != nil {
 log.Fatalf("Failed to connect to cluster: %v", err)
}

// Force a connection to verify our connection string
err = client.Ping(ctx, nil)
if err != nil {
 log.Fatalf("Failed to ping cluster: %v", err)
}

fmt.Println("Connected to DocumentDB!")

collection := client.Database("sample-database").Collection("sample-collection")

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
 log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)
```

```
 ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
 defer cancel()

 cur, err := collection.Find(ctx, bson.D{})

 if err != nil {
  log.Fatalf("Failed to run find query: %v", err)
 }
 defer cur.Close(ctx)

 for cur.Next(ctx) {
  var result bson.M
  err := cur.Decode(&result)
  log.Printf("Returned: %v", result)

  if err != nil {
   log.Fatal(err)
  }
 }

 if err := cur.Err(); err != nil {
  log.Fatal(err)
 }

}

func getCustomTLSConfig(caFile string) (*tls.Config, error) {
 tlsConfig := new(tls.Config)
 certs, err := ioutil.ReadFile(caFile)

 if err != nil {
  return tlsConfig, err
 }

 tlsConfig.RootCAs = x509.NewCertPool()
 ok := tlsConfig.RootCAs.AppendCertsFromPEM(certs)

 if !ok {
  return tlsConfig, errors.New("Failed parsing pem file")
 }

 return tlsConfig, nil
```

Java

Wenn Sie von einer Java-Anwendung aus eine Verbindung zu einem TLS-fähigen Amazon DocumentDB-Cluster herstellen, muss Ihr Programm die von AWS Ihnen bereitgestellte Zertifizierungsstelle (CA) verwenden, um die Verbindung zu validieren. Gehen Sie wie folgt vor, um das Amazon RDS-CA-Zertifikat zu verwenden:

1. Laden Sie die Amazon RDS-CA-Datei von herunter https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem.
2. Erstellen Sie einen Vertrauensspeicher mit dem in der Datei enthaltenen CA-Zertifikat, indem Sie die folgenden Befehle ausführen. Sie müssen den Wert für *<truststorePassword>* ändern. Wenn Sie auf einen Vertrauensspeicher zugreifen, der sowohl das alte CA-Zertifikat (`rds-ca-2015-root.pem`) als auch das neue CA-Zertifikat (`rds-ca-2019-root.pem`) enthält, können Sie das Zertifikat-Bundle in den Vertrauensspeicher importieren.

   Nachfolgend finden Sie ein Beispiel-Shell-Skript, das das Zertifikatpaket in einen Trust Store auf einem Linux-Betriebssystem importiert.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
 {split_after=1}{print > "rds-ca-" n ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "
```

```
keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Es folgt ein Beispiel für ein Shell-Skript, das das Zertifikatspaket in einen Vertrauensspeicher unter macOS importiert.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=<truststorePassword>

curl -sS "https://s3.amazonaws.com/rds-downloads/global-bundle.pem" > ${mydir}/
global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

3. Verwenden Sie das `keystore` in Ihrem Programm, indem Sie die folgenden Systemeigenschaften in Ihrer Anwendung festlegen, bevor Sie eine Verbindung zum Amazon DocumentDB-Cluster herstellen.

```
javax.net.ssl.trustStore: <truststore>
javax.net.ssl.trustStorePassword: <truststorePassword>
```

4. Der folgende Code zeigt, wie Sie mit Java eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
package com.example.documentdb;

import com.mongodb.client.*;
import org.bson.Document;

public final class Test {
    private Test() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
ssl=true&replicaSet=rs0&readpreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
       String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        String truststore = "<truststore>";
        String truststorePassword = "<truststorePassword>";

        System.setProperty("javax.net.ssl.trustStore", truststore);
        System.setProperty("javax.net.ssl.trustStorePassword",
 truststorePassword);

        MongoClient mongoClient = MongoClients.create(connectionString);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection =
 testDB.getCollection("sample-collection");
```

```
        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }

    }
}
```

C# / .NET

Der folgende Code zeigt, wie Sie mit C# /.NET eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace DocDB
{
    class Program
    {
        static void Main(string[] args)
        {
            string template = "mongodb://{0}:{1}@{2}/sampledatabase?
tls=true&replicaSet=rs0&readpreference={3}";
            string username = "<sample-user>";
            string password = "<password>";
            string readPreference = "secondaryPreferred";
```

```
            string clusterEndpoint="sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
            string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);

            string pathToCAFile = "<PATH/global-bundle.pem_file>";

            // ADD CA certificate to local trust store
            // DO this once - Maybe when your service starts
            X509Store localTrustStore = new X509Store(StoreName.Root);
            X509Certificate2Collection certificateCollection = new
 X509Certificate2Collection();
            certificateCollection.Import(pathToCAFile);
            try
            {
                localTrustStore.Open(OpenFlags.ReadWrite);
                localTrustStore.AddRange(certificateCollection);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Root certificate import failed: " + ex.Message);
                throw;
            }
            finally
            {
                localTrustStore.Close();
            }

            var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
            var client = new MongoClient(settings);

            var database = client.GetDatabase("sampledatabase");
            var collection =
 database.GetCollection<BsonDocument>("samplecollection");
            var docToInsert = new BsonDocument { { "pi", 3.14159 } };
            collection.InsertOne(docToInsert);
        }
    }
}
```

mongo shell

Der folgende Code zeigt, wie Sie mit der Mongo-Shell eine Verbindung zu Amazon DocumentDB herstellen und diese abfragen, wenn TLS aktiviert ist.

1. Stellen Sie mit der Mongo-Shell eine Connect zu Amazon DocumentDB her. Wenn Sie eine Mongo-Shell-Version vor 4.2 verwenden, verwenden Sie den folgenden Code, um eine Verbindung herzustellen.

   ```
   mongo --ssl --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --
   sslCAFile global-bundle.pem --username <sample-user> --password <password>
   ```

   Wenn Sie eine Version verwenden, die 4.2 oder höher ist, verwenden Sie den folgenden Code, um eine Verbindung herzustellen. Wiederholbare Schreibvorgänge werden in AWS DocumentDB nicht unterstützt. Ausnahme: Wenn Sie die Mongo-Shell verwenden, fügen Sie den `retryWrites=false` Befehl in keine Codezeichenfolge ein. Standardmäßig sind wiederholbare Schreibvorgänge deaktiviert. `retryWrites=false`Das Einschließen kann zu Fehlern bei normalen Lesebefehlen führen.

   ```
   mongo --tls --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 --
   tlsCAFile global-bundle.pem --username <sample-user> --password <password>
   ```

2. Fügen Sie ein einzelnes Dokument ein.

   ```
   db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
   ```

3. Suchen Sie das Dokument, das zuvor eingefügt wurde.

   ```
   db.myTestCollection.find({'hello':'Amazon DocumentDB'})
   ```

R

Der folgende Code zeigt, wie Sie mit R mithilfe von mongolite (https://jeroen.github.io/mongolite/) eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
#Include the mongolite library.
library(mongolite)
```

```
mongourl <- paste("mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/test2?ssl=true&",
          "readPreference=secondaryPreferred&replicaSet=rs0", sep="")

#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
#   set and specify the read preference as secondary preferred
client <-  mongo(url = mongourl, options = ssl_options(weak_cert_validation = F, ca
 ="<PATH/global-bundle.pem>"))

#Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)

#Find the document that was previously written
client$find()
```

Ruby

Der folgende Code zeigt, wie Sie mit Ruby eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
   database: 'test',
   replica_set: 'rs0',
   read: {:secondary_preferred => 1},
   user: '<sample-user>',
   password: '<password>',
   ssl: true,
   ssl_verify: true,
   ssl_ca_cert: <'PATH/global-bundle.pem'>,
   retry_writes: false
}


begin
   ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
   client = Mongo::Client.new(client_host, client_options)
```

```
   ##Insert a single document
   x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

   ##Find the document that was previously written
   result = client[:test].find()

   #Print the document
   result.each do |document|
      puts JSON.neat_generate(document)
   end
end

#Close the connection
client.close
```

Verbinden bei deaktiviertem TLS

Um ein Codebeispiel für die programmgesteuerte Verbindung zu einem TLS-deaktivierten Amazon DocumentDB-Cluster anzuzeigen, wählen Sie die Registerkarte für die Sprache, die Sie verwenden möchten.

Python

Der folgende Code zeigt, wie Sie mithilfe von Python eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
 and specify the read preference as secondary preferred

import pymongo
import sys

client = pymongo.MongoClient('mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection
```

```
##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
print(x)

##Close the connection
client.close()
```

Node.js

Der folgende Code zeigt, wie Sie mithilfe von Node.js eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
var MongoClient = require('mongodb').MongoClient;

//Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set,
//  and specify the read preference as secondary preferred
var client = MongoClient.connect(
'mongodb://<sample-user>:<password>@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  useNewUrlParser: true
},

function(err, client) {
    if(err)
        throw err;
    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'Amazon DocumentDB'}, function(err, result){
        //Print the result to the screen
```

```
        console.log(result);

        //Close the connection
        client.close()
      });
   });
});
```

PHP

Der folgende Code zeigt, wie Sie mit PHP eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoDB\Client("mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?retryWrites=false");

//Specify the database and collection to be used
$col = $client->sampledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB'] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

Der folgende Code zeigt, wie Sie mit Go eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
package main

import (
 "context"
```

```
 "fmt"
 "log"
 "time"

 "go.mongodb.org/mongo-driver/bson"
 "go.mongodb.org/mongo-driver/mongo"
 "go.mongodb.org/mongo-driver/mongo/options"
)

const (
 // Timeout operations after N seconds
 connectTimeout  = 5
 queryTimeout    = 30
 username        = "<sample-user>"
 password        = "<password>"
 clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

 // Which instances to read from
 readPreference           = "secondaryPreferred"
 connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s"
)

func main() {

 connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
 clusterEndpoint, readPreference)

 client, err := mongo.NewClient(options.Client().ApplyURI(connectionURI))
 if err != nil {
  log.Fatalf("Failed to create client: %v", err)
 }

 ctx, cancel := context.WithTimeout(context.Background(),
 connectTimeout*time.Second)
 defer cancel()

 err = client.Connect(ctx)
 if err != nil {
  log.Fatalf("Failed to connect to cluster: %v", err)
 }

 // Force a connection to verify our connection string
 err = client.Ping(ctx, nil)
```

```
if err != nil {
 log.Fatalf("Failed to ping cluster: %v", err)
}

fmt.Println("Connected to DocumentDB!")

collection := client.Database("sample-database").Collection("sample-collection")

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
 log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

cur, err := collection.Find(ctx, bson.D{})

if err != nil {
 log.Fatalf("Failed to run find query: %v", err)
}
defer cur.Close(ctx)

for cur.Next(ctx) {
 var result bson.M
 err := cur.Decode(&result)
 log.Printf("Returned: %v", result)

 if err != nil {
  log.Fatal(err)
 }
}

if err := cur.Err(); err != nil {
 log.Fatal(err)
}
```

```
}
```

Java

Der folgende Code zeigt, wie Sie mit Java eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
package com.example.documentdb;

import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.ServerAddress;
import com.mongodb.MongoException;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoCollection;
import org.bson.Document;


public final class Main {
    private Main() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s";
        String username = "<sample-user>";
        String password = "<password>";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        MongoClientURI clientURI = new MongoClientURI(connectionString);
        MongoClient mongoClient = new MongoClient(clientURI);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection = testDB.getCollection("sample-
collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);
```

```
        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }

    }
}
```

C# / .NET

Der folgende Code zeigt, wie Sie mit C# /.NET eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace CSharpSample
{
   class Program
    {
       static void Main(string[] args)
        {
           string template = "mongodb://{0}:{1}@{2}/sampledatabase?
replicaSet=rs0&readpreference={3}";
           string username = "<sample-user>";
           string password = "<password>";
           string clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
           string readPreference = "secondaryPreferred";
           string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);
```

```
            var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
            var client = new MongoClient(settings);

            var database = client.GetDatabase("sampledatabase");
            var collection =
 database.GetCollection<BsonDocument>("samplecollection");
            var docToInsert = new BsonDocument { { "pi", 3.14159 } };
             collection.InsertOne(docToInsert);
        }
    }
}
```

mongo shell

Der folgende Code zeigt, wie Sie mit der Mongo-Shell eine Verbindung zu Amazon DocumentDB herstellen und diese abfragen, wenn TLS deaktiviert ist.

1. Stellen Sie mit der Mongo-Shell eine Connect zu Amazon DocumentDB her.

   ```
   mongo --host mycluster.node.us-east-1.docdb.amazonaws.com:27017 --
   username <sample-user> --password <password>
   ```

2. Fügen Sie ein einzelnes Dokument ein.

   ```
   db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
   ```

3. Suchen Sie das Dokument, das zuvor eingefügt wurde.

   ```
   db.myTestCollection.find({'hello':'Amazon DocumentDB'})
   ```

R

Der folgende Code zeigt, wie Sie mit R mithilfe von mongolite (https://jeroen.github.io/mongolite/) eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
#Include the mongolite library.
library(mongolite)

#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
```

```
#   set and specify the read preference as secondary preferred
client <- mongo(url = "mongodb://<sample-user>:<password>@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/sample-database?
readPreference=secondaryPreferred&replicaSet=rs0")

##Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)

##Find the document that was previously written
client$find()
```

Ruby

Der folgende Code zeigt, wie Sie mit Ruby eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
   database: 'test',
   replica_set: 'rs0',
   read: {:secondary_preferred => 1},
   user: '<sample-user>',
   password: '<password>',
   retry_writes: false
}

begin
   ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
   client = Mongo::Client.new(client_host, client_options)

   ##Insert a single document
   x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

   ##Find the document that was previously written
   result = client[:test].find()

   #Print the document
   result.each do |document|
```

```
      puts JSON.neat_generate(document)
   end
end

#Close the connection
client.close
```

Change Streams mit Amazon DocumentDB verwenden

Die Change-Streams-Funktion in Amazon DocumentDB (mit MongoDB-Kompatibilität) bietet eine zeitlich geordnete Abfolge von Änderungsereignissen, die in den Sammlungen Ihres Clusters auftreten. Sie können Ereignisse aus einem Change Stream lesen, um zahlreiche verschiedene Anwendungsfälle zu implementieren, einschließlich:

- Änderungsbenachrichtigung
- Volltextsuche mit Amazon OpenSearch Dienst (OpenSearch Bedienung)
- Analytik mit Amazon Redshift

Anwendungen können Änderungsstreams verwenden, um Datenveränderungen bei individuellen Sammlungen zu abonnieren. Die Ereignisse werden in Change Streams in der Reihenfolge angeordnet, wie sie im Cluster auftreten, und nach der Aufzeichnung des Ereignisses 3 Stunden (Standardeinstellung) gespeichert. Die Aufbewahrungsfrist kann auf bis zu 7 Tage verlängert werden, indem Sie den`change_stream_log_retention_duration`Parameter. Informationen zum Ändern der Aufbewahrungsfrist für Change-Streams finden Sie unterÄnderung der Aufbewahrungsdauer für das Change-Stream-Protokoll.

Themen

- Unterstützte -Vorgänge
- Fakturierung
- Einschränkungen
- Aktivieren von Change Streams
- Beispiel: Verwendung von Change Streams mit Python
- Vollständige Dokumentsuche
- Wiederaufnahme eines Change Streams
- Einen Change-Stream fortsetzen mitstartAtOperationTime

- Transaktionen in Change-Streams
- Ändern des Aufbewahrungszeitraums für das Change Stream-Protokoll

Unterstützte -Vorgänge

Amazon DocumentDB unterstützt die folgenden Operationen für Change-Streams:

- Alle Änderungsereignisse werden in der MongoDB unterstütztdb.collection.watch(),db.watch()undclient.watch()API.
- Vollständige Dokumentsuche nach Aktualisierungen.
- Aggregationsstufen:$match,$project,$redact, und$addFieldsund$replaceRoot.
- Einen Change-Stream von einem Resume-Token aus fortsetzen
- Wiederaufnahme eines Change-Streams von einem Zeitstempel aus mitstartAtOperation(gilt für Amazon DocumentDB v4.0+)

Fakturierung

Die Amazon DocumentDB-Funktion zum Ändern von Streams ist standardmäßig deaktiviert und es fallen keine zusätzlichen Gebühren an, bis die Funktion aktiviert ist. Die Verwendung von Change-Streams in einem Cluster verursacht zusätzliche Lese- und Schreib-IOs sowie Speicherkosten. Sie können das verwendenmodifyChangeStreamsAPI-Vorgang, um diese Funktion für Ihren Cluster zu aktivieren. Weitere Informationen zur Preisgestaltung finden Sie unterPreise für Amazon DocumentDB.

Einschränkungen

Für Change-Streams gelten in Amazon DocumentDB die folgenden Einschränkungen:

- Change-Streams können nur über eine Verbindung zur primären Instance eines Amazon DocumentDB-Clusters geöffnet werden. Das Lesen von Change Streams auf einer Replikat-Instance wird derzeit nicht unterstützt. Beim Aufruf der API-Operation `watch()` müssen Sie die Leseeinstellung **`primary`** angeben, um sicherzustellen, dass alle Lesevorgänge an die primäre Instance weitergeleitet werden (siehe den Abschnitt Beispiel).
- Ereignisse, die für eine Sammlung in einen Änderungsstream geschrieben werden, sind bis zu 7 Tage lang verfügbar (die Standardeinstellung ist 3 Stunden). Änderungsstream-Daten werden nach

dem Zeitfenster der Protokollaufbewahrungsdauer gelöscht, auch wenn keine neuen Änderungen vorgenommen wurden.

- Eine über längere Zeit für eine Sammlung ausgeführte Schreiboperation wie `updateMany` oder `deleteMany` kann das Schreiben von Change Stream-Ereignissen vorübergehend bis zum Abschluss der über längere Zeit ausgeführten Schreiboperation blockieren.
- Amazon DocumentDB unterstützt das MongoDB-Betriebsprotokoll nicht (`oplog`).
- Bei Amazon DocumentDB müssen Sie Change-Streams für eine bestimmte Sammlung explizit aktivieren.
- Wenn die Gesamtgröße eines Change Stream-Ereignisses (einschließlich der Änderungsdaten und des vollständigen Dokuments, wenn angefordert) größer als `16 MB` ist, tritt auf dem Client ein Lesefehler für die Change Streams auf.
- Der Ruby-Treiber wird derzeit nicht unterstützt, wenn`db.watch()`und`client.watch()`mit Amazon DocumentDB v3.6.

Aktivieren von Change Streams

Sie können Amazon DocumentDB-Change-Streams für alle Sammlungen innerhalb einer bestimmten Datenbank oder nur für ausgewählte Sammlungen aktivieren. Im Folgenden finden Sie Beispiele für die Aktivierung von Change Streams für verschiedene Anwendungsfälle über die Mongo-Shell. Leere Zeichenfolgen werden bei der Angabe von Datenbank- und Sammlungsnamen als Platzhalter behandelt.

```
//Enable change streams for the collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "foo",
    enable: true});
```

```
//Disable change streams on collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "foo",
    enable: false});
```

```
//Enable change streams for all collections in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
```

```
    collection: "",
    enable: true});
```

```
//Enable change streams for all collections in all databases in a cluster
db.adminCommand({modifyChangeStreams: 1,
    database: "",
    collection: "",
    enable: true});
```

Change Streams werden für eine Sammlung aktiviert, wenn eine der folgenden Bedingungen erfüllt ist:

- Sowohl die Datenbank als auch die Sammlung sind explizit aktiviert.
- Die Datenbank, die die Sammlung enthält, ist aktiviert.
- Alle Datenbanken sind aktiviert.

Wenn Sie eine Sammlung aus einer Datenbank löschen, werden Change Streams für diese Sammlung nicht deaktiviert, wenn Change Streams für die übergeordnete Datenbank auch aktiviert sind oder wenn alle Datenbanken im Cluster aktiviert sind. Wenn eine neue Sammlung mit demselben Namen wie die gelöschte Sammlung erstellt wird, werden Change Streams für diese Sammlung aktiviert.

Sie können alle aktivierten Change Streams Ihres Clusters mithilfe der `$listChangeStreams`-Aggregationspipeline-Phase auflisten. Alle von Amazon DocumentDB unterstützten Aggregationsphasen können in der Pipeline für zusätzliche Verarbeitungsvorgänge verwendet werden. Wenn eine zuvor aktivierte Sammlung deaktiviert wurde, wird sie nicht in der `$listChangeStreams`-Ausgabe angezeigt.

```
//List all databases and collections with change streams enabled
cursor = new DBCommandCursor(db,
    db.runCommand(
        {aggregate: 1,
        pipeline: [{$listChangeStreams: 1}],
        cursor:{}}));
```

```
//List of all databases and collections with change streams enabled
{ "database" : "test", "collection" : "foo" }
{ "database" : "bar", "collection" : "" }
```

```
{ "database" : "", "collection" : "" }
```

```
//Determine if the database “bar” or collection “bar.foo” have change streams enabled
cursor = new DBCommandCursor(db,
  db.runCommand(
      {aggregate: 1,
       pipeline: [{$listChangeStreams: 1},
                  {$match: {$or: [{database: "bar", collection: "foo"},
                                  {database: "bar", collection: ""},
                                  {database: "", collection: ""}]}}
                ],
      cursor:{}}));
```

Beispiel: Verwendung von Change Streams mit Python

Im Folgenden finden Sie ein Beispiel für die Verwendung eines Amazon DocumentDB-Change-Streams mit Python auf Sammlungsebene.

```
import os
import sys
from pymongo import MongoClient, ReadPreference

username = "DocumentDBusername"
password = <Insert your password>

clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']

#While ‘Primary’ is the default read preference, here we give an example of
#how to specify the required read preference when reading the change streams
coll = db.get_collection('foo', read_preference=ReadPreference.PRIMARY)
#Create a stream object
stream = coll.watch()
#Write a new document to the collection to generate a change event
coll.insert_one({'x': 1})
#Read the next change event from the stream (if any)
print(stream.try_next())

"""
```

```
Expected Output:
{'_id': {'_data': '015daf94f6000000020100000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""

#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
Expected Output:
None
"""

#Generate a new change event by updating a document
result = coll.update_one({'x': 1}, {'$set': {'x': 2}})
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf99d4000000010100000100009025'},
'clusterTime': Timestamp(1571789268, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 2}}}
"""
```

Das Folgende ist ein Beispiel für die Verwendung eines Amazon DocumentDB-Change-Streams mit Python auf Datenbankebene.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
```

```
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']
#Create a stream object
stream = db.watch()
coll = db.get_collection('foo')
#Write a new document to the collection foo to generate a change event
coll.insert_one({'x': 1})

#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""
#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
Expected Output:
None
"""

coll = db.get_collection('foo1')

#Write a new document to another collection to generate a change event
coll.insert_one({'x': 1})
print(stream.try_next())

"""
Expected Output: Since the change stream cursor was the database level you can see
 change events from different collections in the same database
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
```

```
'ns': {'coll': 'foo1', 'db': 'bar'},
'operationType': 'insert'}
"""
```

Vollständige Dokumentsuche

Das Änderungsereignis „Aktualisierung“ enthält nicht das vollständige Dokument, sondern lediglich die ausgeführte Änderung. Wenn für Ihren Anwendungsfall das vollständige Dokument erforderlich ist, das von einer Aktualisierung betroffen ist, können Sie beim Öffnen des Datenstroms die vollständige Dokumentsuche aktivieren.

Das `fullDocument`-Dokument für ein Update-Change Stream-Ereignis stellt die neueste Version des aktualisierten Dokuments zum Zeitpunkt der Dokumentsuche dar. Wenn zwischen der Aktualisierungsoperation und der `fullDocument`-Suche Änderungen ausgeführt wurden, besitzt das `fullDocument`-Dokument möglicherweise nicht den Dokumentstatus zur Aktualisierungszeit.

```
#Create a stream object with update lookup enabled
stream = coll.watch(full_document='updateLookup')

#Generate a new change event by updating a document
result = coll.update_one({'x': 2}, {'$set': {'x': 3}})

stream.try_next()

#Output:
{'_id': {'_data': '015daf9b7c000000010100000100009025'},
'clusterTime': Timestamp(1571789692, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'fullDocument': {'_id': ObjectId('5daf9502ea258751778163d7'), 'x': 3},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 3}}}
```

Wiederaufnahme eines Change Streams

Sie können einen Change Stream zu einem späteren Zeitpunkt mithilfe eines Fortsetzungs-Tokens fortsetzen, das dem Feld `_id` des zuletzt abgerufenen Änderungsereignisdokuments entspricht.

```
import os
import sys
from pymongo import MongoClient
```

```
username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem', retryWrites='false')

db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
token = event['_id']
print(token)

"""
Output: This is the resume token that we will later us to resume the change stream
{'_data': '015daf9c5b0000000101000000100009025'}
"""
#Python provides a nice shortcut for getting a stream’s resume token
print(stream.resume_token)

"""
Output
{'_data': '015daf9c5b0000000101000000100009025'}
"""
#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
#Generate another change event by inserting a document
result = coll.insert_one({'y': 5})
#Open a stream starting after the selected resume token
stream = db.watch(full_document='updateLookup', resume_after=token)
#Our first change event is the update with the specified _id
print(stream.try_next())

"""
#Output: Since we are resuming the change stream from the resume token, we will see all
 events after the first update operation. In our case, the change stream will resume
 from the update operation {x:5}

{'_id': {'_data': '015f7e8f0c0000000601000000006000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602129676, 6),
```

```
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0ac423bafbfd9adba2')},
'fullDocument': {'_id': ObjectId('5f7e8f0ac423bafbfd9adba2'), 'x': 5},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""
#Followed by the insert
print(stream.try_next())

"""
#Output:
{'_id': {'_data': '015f7e8f0c0000000701000000007000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602129676, 7),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94')},
'fullDocument': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94'), 'y': 5}}
"""
```

Einen Change-Stream fortsetzen mit`startAtOperationTime`

Sie können einen Change-Stream zu einem späteren Zeitpunkt ab einem bestimmten Zeitstempel wieder aufnehmen, indem Sie`startAtOperationTime`.

> **Note**
>
> Die Fähigkeit zu verwenden`startAtOperationTime`ist in Amazon DocumentDB 4.0+ verfügbar. Bei der Verwendung`startAtOperationTime`, gibt der Change-Stream-Cursor nur Änderungen zurück, die zu oder nach dem angegebenen Zeitstempel aufgetreten sind. Der`startAtOperationTime`und`resumeAfter`Befehle schließen sich gegenseitig aus und können daher nicht zusammen verwendet werden.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='rds-root-ca-2020.pem',retryWrites='false')
```

```
db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
timestamp = event['clusterTime']
print(timestamp)
"""
Output
Timestamp(1602129114, 4)
"""
#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
result = coll.insert_one({'y': 5})
#Generate another change event by inserting a document
#Open a stream starting after specified time stamp

stream = db.watch(start_at_operation_time=timestamp)
print(stream.try_next())

"""
#Output: Since we are resuming the change stream at the time stamp of our first update
 operation (x:4), the change stream cursor will point to that event
{'_id': {'_data': '015f7e941a0000000301000000003000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602130970, 3),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e9417c423bafbfd9adbb1')},
'updateDescription': {'updatedFields': {'x': 4}, 'removedFields': []}}
"""

print(stream.try_next())
"""
#Output: The second event will be the subsequent update operation (x:5)
{'_id': {'_data': '015f7e9502000000050100000005000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602131202, 5),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e94ffc423bafbfd9adbb2')},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""

print(stream.try_next())
```

```
"""
#Output: And finally the last event will be the insert operation (y:5)
{'_id': {'_data': '015f7e9502000000060100000006000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602131202, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e95025c4a569e0f6dde92')},
'fullDocument': {'_id': ObjectId('5f7e95025c4a569e0f6dde92'), 'y': 5}}
"""
```

Transaktionen in Change-Streams

Change-Stream-Ereignisse enthalten keine Ereignisse, die auf Transaktionen zurückzuführen sind, für die kein Commitment abgeschlossen wurde und/oder abgebrochen wurde. Zum Beispiel, wenn Sie eine Transaktion mit einer beginnen`INSERT`Operation und eins`UPDATE`Operation und. Wenn dein`INSERT`Operation ist erfolgreich, aber die`UPDATE`Der Vorgang schlägt fehl, die Transaktion wird zurückgesetzt. Da diese Transaktion zurückgesetzt wurde, enthält Ihr Change-Stream keine Ereignisse für diese Transaktion.

Ändern des Aufbewahrungszeitraums für das Change Stream-Protokoll

Sie können die Aufbewahrungsdauer des Change-Stream-Protokolls auf einen Wert zwischen 1 Stunde und 7 Tagen ändern, indem SieAWS Management Consoleoder dieAWS CLI.

Using the AWS Management Console

So ändern Sie den Aufbewahrungszeitraum für das Change-Stream-Protokoll

1. Melden Sie sich an beiAWS Management Console, und öffnen Sie die Amazon DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parameter groups (Parametergruppen) die Cluster-Parametergruppe, die Ihrem Cluster zugeordnet ist. Informationen zum Identifizieren der Clusterparametergruppe, die dem Cluster zugeordnet ist, finden Sie unter Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters.
4. Auf der resultierenden Seite werden die Parameter und die entsprechenden Details zu Ihrer Cluster-Parametergruppe angezeigt. Wählen Sie den `change_stream_log_retention_duration`-Parameter aus.
5. Klicken Sie oben rechts auf der Seite auf Edit (Bearbeiten), um den Wert des Parameters zu ändern. Die`change_stream_log_retention_duration`Der Parameter kann so geändert werden, dass er zwischen 1 Stunde und 7 Tagen liegt.
6. Führen Sie die Änderung aus. Wählen Sie anschließend Modify cluster parameter (Cluster-Parameter ändern) aus, um die Änderungen zu speichern. Um die Änderungen zu verwerfen, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Um den `change_stream_log_retention_duration`-Parameter Ihrer Cluster-Parametergruppe zu ändern, verwenden Sie die `modify-db-cluster-parameter-group`-Operation mit den folgenden Parametern:

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die Sie ändern. Informationen zum Identifizieren der Clusterparametergruppe, die dem Cluster zugeordnet ist, finden Sie unter Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters.
- **`--parameters`** – Erforderlich. Der Parameter, der von Ihnen geändert wird. Jeder Parametereintrag muss Folgendes enthalten:
  - **`ParameterName`**— Der Name des Parameters, den Sie ändern. In diesem Fall ist es `change_stream_log_retention_duration`
  - **`ParameterValue`**— Der neue Wert für diesen Parameter.
  - **`ApplyMethod`**— Wie die Änderungen an diesem Parameter angewendet werden sollen. Zugelassene Werte sind `immediate` und `pending-reboot`.

> **Note**
>
> Parameter mit dem `ApplyType` von `static` müssen über einen `ApplyMethod` von `pending-reboot` verfügen.

1. Um die Werte des Parameters `change_stream_log_retention_duration` zu ändern, führen Sie den folgenden Befehl aus und ersetzen `parameter-value` durch den Wert, auf den Sie den Parameter ändern möchten.

   Für Linux, macOS oder Unix:

   ```
   aws docdb modify-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --parameters
    "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-
   value>,ApplyMethod=immediate"
   ```

   Für Windows:

   ```
   aws docdb modify-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --parameters
    "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-
   value>,ApplyMethod=immediate"
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

   ```
   {
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
   ```

2. Warten Sie mindestens 5 Minuten.
3. Listen Sie die Parameterwerte von `sample-parameter-group` auf, um sicherzustellen, dass Ihre Änderungen übernommen wurden.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-cluster-parameters \
   ```

```
    --db-cluster-parameter-group-name sample-parameter-group
```

Für Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "Parameters": [
        {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "change_stream_log_retention_duration",
            "ParameterValue": "12345",
            "Description": "Duration of time in seconds that the change stream
 log is retained and can be consumed.",
            "Source": "user",
            "ApplyType": "dynamic",
            "DataType": "integer",
            "AllowedValues": "3600-86400",
            "IsModifiable": true,
            "ApplyMethod": "immediate"
        }
    ]
}
```

> **Note**
>
> Bei der Aufbewahrung von Change-Stream-Protokollen werden keine Protokolle gelöscht, die älter sind als die konfigurierten`change_stream_log_retention_duration`Wert, bis die Protokollgröße größer als (>) 51.200 MB ist.

VerwendenAWS Lambdamit Change Streams

Amazon DocumentDB ist integriert in`AWS Lambda`sodass Sie Lambda-Funktionen verwenden können, um Datensätze in einem Change-Stream zu verarbeiten. Die Zuordnung von Lambda-Ereignisquellen ist eine Ressource, mit der Lambda-Funktionen aufgerufen werden können, um Amazon DocumentDB-Ereignisse zu verarbeiten, die Lambda nicht direkt aufrufen. Mit Amazon DocumentDB Change Stream als Ereignisquelle können Sie ereignisgesteuerte Anwendungen erstellen, die auf Änderungen in Ihren Daten reagieren. Beispielsweise können Sie Lambda-Funktionen verwenden, um neue Dokumente zu verarbeiten, Aktualisierungen vorhandener Dokumente nachzuverfolgen oder gelöschte Dokumente zu protokollieren.

Sie können eine Ereignisquellenzuordnung so konfigurieren, dass Datensätze aus Ihrem Amazon DocumentDB-Change-Stream an eine Lambda-Funktion gesendet werden. Ereignisse können einzeln oder gebündelt gesendet werden, um die Effizienz zu verbessern, und sie werden der Reihe nach verarbeitet. Sie können das Batchverhalten Ihrer Ereignisquellenzuordnung auf der Grundlage einer bestimmten Zeitfensterdauer (0-300 Sekunden) oder der Anzahl von Batch-Datensätzen (maximale Grenze von 10.000 Datensätzen) konfigurieren. Sie können mehrere Zuordnungen von Ereignisquellen erstellen, um dieselben Daten mit mehreren Lambda-Funktionen zu verarbeiten oder um unterschiedliche Elemente aus mehreren Streams mit einer einzigen Funktion zu verarbeiten.

Wenn Ihre Funktion einen Fehler zurückgibt, versucht Lambda den Batch erneut, bis er erfolgreich verarbeitet wurde. Falls die Ereignisse im Change-Stream abgelaufen sind, deaktiviert Lambda die Zuordnung der Ereignisquellen. In diesem Fall können Sie eine neue Zuordnung der Ereignisquelle erstellen und diese mit einer Startposition Ihrer Wahl konfigurieren. Lambda-Ereignisquellenzuordnungen verarbeiten Ereignisse aufgrund der verteilten Natur ihrer Poller mindestens einmal. Infolgedessen kann Ihre Lambda-Funktion in seltenen Situationen doppelte Ereignisse erhalten. Folgen Sie den bewährten Methoden für die Arbeit mit`AWS Lambda`Funktionen und erstellen idempotente Funktionen, um Probleme im Zusammenhang mit doppelten Ereignissen zu vermeiden. Weitere Informationen finden Sie unter[VerwendenAWS Lambda consolemit Amazon DocumentDB](#)in der`AWS Lambda`Leitfaden für Entwickler.

Gemäß bewährten Methoden für die Leistung muss die Lambda-Funktion kurzlebig sein. Um unnötige Verarbeitungsverzögerungen zu vermeiden, sollte sie auch keine komplexe Logik ausführen. Insbesondere bei einem Hochgeschwindigkeits-Stream ist es besser, asynchrone Nachbearbeitungs-Schrittfunktions-Workflows auszulösen als synchrone Lambdas mit langer Laufzeit. Weitere Informationen über AWS Lambda finden Sie im AWS Lambda-Entwicklerleitfaden.

Einschränkungen

Die folgenden Einschränkungen sind bei der Arbeit mit Amazon DocumentDB zu beachten undAWS Lambda:

- AWS Lambdawird derzeit nur auf Amazon DocumentDB 4.0 und 5.0 unterstützt.
- AWS Lambdawird derzeit nicht auf elastischen Clustern oder globalen Clustern unterstützt.
- AWS LambdaDie Größe der Nutzlast darf 6 MB nicht überschreiten. Weitere Informationen zu Lambda-Batchgrößen finden Sie unter „Batching-Verhalten“ inZuordnungen von Lambda-EreignisquellenAbschnitt in derAWS LambdaLeitfaden für Entwickler.

Verwenden der JSON-Schemavalidierung

Mit dem `$jsonSchema` Auswertungsabfrage-Operator können Sie überprüfen, ob Dokumente in Ihre Sammlungen eingefügt werden.

Themen

- Erstellen und Verwenden der JSON-Schemavalidierung
- Unterstützte Schlüsselwörter
- Einschränkungen

Erstellen und Verwenden der JSON-Schemavalidierung

Erstellen einer Sammlung mit Schemavalidierung

Sie können eine Sammlung mit `createCollection` Betriebs- und Validierungsregeln erstellen. Diese Validierungsregeln werden bei Einfügungen oder Aktualisierungen von Amazon DocumentDB-Dokumenten angewendet. Das folgende Codebeispiel zeigt Validierungsregeln für eine Sammlung von Mitarbeitern:

```
db.createCollection("employees", {
   "validator": {
      "$jsonSchema": {
         "bsonType": "object",
         "title": "employee validation",
         "required": [ "name", "employeeId"],
         "properties": {
            "name": {
                  "bsonType": "object",
                  "properties": {
                     "firstName": {
                        "bsonType": ["string"]
                     },
                     "lastName": {
                        "bsonType": ["string"]
                     }
                  },
                  "additionalProperties" : false
            },
            "employeeId": {
               "bsonType": "string",
               "description": "Unique Identifier for employee"
            },
             "salary": {
               "bsonType": "double"
            },
            "age": {
               "bsonType": "number"
            }
         },
         "additionalProperties" : true
      }
   },
   "validationLevel": "strict", "validationAction": "error"
} )
```

Einfügen eines gültigen Dokuments

Im folgenden Beispiel werden Dokumente eingefügt, die den oben genannten Schemavalidierungsregeln entsprechen:

```
db.employees.insert({"name" : { "firstName" : "Carol" , "lastName" : "Smith"},
 "employeeId": "c720a" , "salary": 1000.0 })
```

```
db.employees.insert({ "name" : { "firstName" : "William", "lastName" : "Taylor" },
 "employeeId" : "c721a", "age" : 24})
```

Einfügen eines ungültigen Dokuments

Im folgenden Beispiel werden Dokumente eingefügt, die nicht den oben genannten Schemavalidierungsregeln entsprechen. In diesem Beispiel ist der Wert employeeId keine Zeichenfolge:

```
db.employees.insert({
    "name" : { "firstName" : "Carol" , "lastName" : "Smith"},
    "employeeId": 720 ,
    "salary": 1000.0
})
```

Dieses Beispiel zeigt eine falsche Syntax innerhalb des -Dokuments.

Ändern einer Sammlung

Der `collMod` Befehl wird verwendet, um Validierungsregeln für vorhandene Sammlungen hinzuzufügen oder zu ändern. Im folgenden Beispiel wird der erforderlichen Feldliste ein Trichterfeld hinzugefügt:

```
db.runCommand({"collMod" : "employees",
   "validator": {
      "$jsonSchema": {
         "bsonType": "object",
         "title": "employee validation",
         "required": [ "name", "employeeId", "salary"],
         "properties": {
            "name": {
                  "bsonType": "object",
                  "properties": {
                     "firstName": {
                        "bsonType": ["string"]
                     },
                     "lastName": {
                        "bsonType": ["string"]
                     }
                  },
                  "additionalProperties" : false
```

```
            },
            "employeeId": {
               "bsonType": "string",
               "description": "Unique Identifier for employee"
            },
             "salary": {
               "bsonType": "double"
            },
            "age": {
               "bsonType": "number"
            }
         },
         "additionalProperties" : true
      }
   }
} )
```

Umgang mit Dokumenten, die hinzugefügt wurden, bevor die Validierungsregeln geändert wurden

Verwenden Sie die folgenden `validationLevel` Modifikatoren, um Dokumente zu adressieren, die Ihrer Sammlung hinzugefügt wurden, bevor die Validierungsregeln geändert wurden:

- Strict : Wendet Validierungsregeln auf alle Einfügungen und Aktualisierungen an.
- moderiert : Wendet Validierungsregeln auf vorhandene gültige Dokumente an. Während Updates werden vorhandene ungültige Dokumente nicht überprüft.

Im folgenden Beispiel ist nach der Aktualisierung der Validierungsregeln für die Sammlung mit dem Namen „Mitarbeiter“ das Suchfeld erforderlich. Das Aktualisieren des folgenden Dokuments schlägt fehl:

```
db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
"n" : 0,
    "nModified" : 0,
    "writeErrors" : [
        {
"index" : 0,
            "code" : 121,
            "errmsg" : "Document failed validation"
        }
    ],
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Wenn Sie die Validierungsebene auf aktualisieren`moderate`, kann das obige Dokument erfolgreich aktualisiert werden:

```
db.runCommand({
    "collMod" : "employees",
    validationLevel : "moderate"
})

db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
"n" : 1,
    "nModified" : 1,
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Abrufen von Dokumenten mit dem $jsonSchema

Der `$jsonSchema` Operator kann als Filter verwendet werden, um Dokumente abzufragen, die dem JSON-Schema entsprechen. Dies ist ein Operator der obersten Ebene, der in Filterdokumenten als Feld der obersten Ebene vorhanden sein oder mit Abfrageoperatoren wie `$and`, `$or`und verwendet werden kann`$nor`. Die folgenden Beispiele zeigen die Verwendung von $jsonSchema als individuellen Filter und mit anderen Filteroperatoren:

In eine „Mitarbeiter“-Sammlung eingefügtes Dokument:

```
{ "name" : { "firstName" : "Carol", "lastName" : "Smith" }, "employeeId" : "c720a",
 "salary" : 1000 }
{ "name" : { "firstName" : "Emily", "lastName" : "Brown" }, "employeeId" : "c720b",
 "age" : 25, "salary" : 1050.2 }
{ "name" : { "firstName" : "William", "lastName" : "Taylor" }, "employeeId" : "c721a",
 "age" : 24, "salary" : 1400.5 }
{ "name" : { "firstName" : "Jane", "lastName" : "Doe" }, "employeeId" : "c721a",
 "salary" : 1300 }
```

Die Sammlung wurde nur mit dem `$jsonSchema` Operator gefiltert:

```
db.employees.find({
       $jsonSchema: { required: ["age"] } })
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
 "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
 "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

Sammlung gefiltert mit dem `$jsonSchema` Operator und einem anderen Operator:

```
db.employees.find({
       $or: [{ $jsonSchema: { required: ["age", "name"]}},
            { salary: { $lte:1000}}]});
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{ "_id" : ObjectId("64e5f8886218c620cf0e8f8a"), "name" : { "firstName" : "Carol",
 "lastName" : "Smith" }, "employeeId" : "c720a", "salary" : 1000 }
```

```
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
 "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
 "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

Sammlung gefiltert mit dem `$jsonSchema` Operator und mit `$match` im Aggregatfilter:

```
db.employees.aggregate(
    [{ $match: {
        $jsonSchema: {
            required: ["name", "employeeId"],
            properties: {"salary" :{"bsonType": "double"}}
        }
       }
    }]
)
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
"_id" : ObjectId("64e5f8886218c620cf0e8f8a"),
 "name" : { "firstName" : "Carol", "lastName" : "Smith" },
"employeeId" : "c720a",
"salary" : 1000
}
{
"_id" : ObjectId("64e5f91c6218c620cf0e8f8b"),
"name" : { "firstName" : "Emily", "lastName" : "Brown" },
"employeeId" : "c720b",
"age" : 25,
"salary" : 1050.2
}
{
"_id" : ObjectId("64e5f94e6218c620cf0e8f8c"),
"name" : { "firstName" : "William", "lastName" : "Taylor" },
"employeeId" : "c721a",
"age" : 24,
"salary" : 1400.5
}
{
"_id" : ObjectId("64e5f9786218c620cf0e8f8d"),
"name" : { "firstName" : "Jane", "lastName" : "Doe" },
"employeeId" : "c721a",
```

```
"salary" : 1300
}
```

Anzeigen vorhandener Validierungsregeln

Um die vorhandenen Validierungsregeln für eine Sammlung anzuzeigen, verwenden Sie:

```
db.runCommand({
    listCollections: 1,
    filter: { name: 'employees' }
})
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "name" : "employees",
                "type" : "collection",
                "options" : {
                    "autoIndexId" : true,
                    "capped" : false,
                    "validator" : {
                        "$jsonSchema" : {
                            "bsonType" : "object",
                            "title" : "employee validation",
                            "required" : [
                                "name",
                                "employeeId",
                                "salary"
                            ],
                            "properties" : {
                                "name" : {
                                    "bsonType" : "object",
                                    "properties" : {
                                        "firstName" : {
                                            "bsonType" : [
                                                "string"
                                            ]
                                        },
                                        "lastName" : {
```

```
                                        "bsonType" : [
                                            "string"
                                        ]
                                    }
                                },
                                "additionalProperties" : false
                            },
                            "employeeId" : {
                                "bsonType" : "string",
                                "description" : "Unique Identifier for employee"
                            },
                            "salary" : {
                                "bsonType" : "double"
                            },
                            "age" : {
                                "bsonType" : "number"
                            }
                        },
                        "additionalProperties" : true
                    }
                },
                "validationLevel" : "moderate",
                "validationAction" : "error"
            },
            "info" : {
                "readOnly" : false
            },
            "idIndex" : {
                "v" : 2,
                "key" : {
                    "_id" : 1
                },
                "name" : "_id_",
                "ns" : "test.employees"
            }
        }
    ],
    "id" : NumberLong(0),
    "ns" : "test.$cmd.listCollections"
},
"ok" : 1,
"operationTime" : Timestamp(1692788937, 1)
}
```

Unterstützte Schlüsselwörter

Die folgenden Felder werden in den `collMod` Befehlen `create` und unterstützt:

- **`Validator`** – Unterstützt einen `$jsonSchem`Operator.
- **`ValidationLevel`** – Unterstützt die `moderate` Werte `strict`, und `off`.
- **`ValidationAction`** – Unterstützt den -`error`Wert.

Der Operator $jsonSchema unterstützt die folgenden Schlüsselwörter:

- `additionalItems`
- `additionalProperties`
- `allOf`
- `anyOf`
- `bsonType`
- `dependencies`
- `description`
- `enum`
- `exclusiveMaximum`
- `exclusiveMinimum`
- `items`
- `maximum`
- `minimum`
- `maxItems`
- `minItems`
- `maxLength`
- `minLength`
- `maxProperties`
- `minProperties`
- `multipleOf`
- `not`
- `oneOf`

- pattern
- patternProperties
- properties
- required
- title
- type
- uniqueItems

Einschränkungen

Die folgenden Einschränkungen gelten für die $jsonSchema Validierung:

- Sammlungen behalten keine Validierungsregeln bei, wenn Schreibvorgänge mit $out Aggregationsstufe in die Sammlung geschrieben werden. Um gültige Dokumente in der Ausgabesammlung abzurufen, empfehlen wir, den $jsonSchema Filter in der $match Aggregatphase vor der $out Aggregationsphase zu verwenden. Die Schemavalidierung kann nach der $outAggregationsphase erneut auf die Ausgabesammlung angewendet werden.

  ```
  db.foo.aggregate([{$match: {$jsonSchema: {...}}}, {$out: "bar"}]);
  db.runCommand("collMod": "bar", validator: {$jsonSchema: {...}})
  ```

- Amazon DocumentDB gibt den Fehler „Dokumentieren fehlgeschlagener Validierung“ zurück, wenn eine Operation die Validierungsregel nicht besteht.
- Amazon DocumentDB unterstützt keine bypassDocumentValidation Aktion in db.runCommand Betrieb.
- Elastische Amazon DocumentDB-Cluster unterstützen nicht$jsonSchema.

Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz

Wenn Sie gegen Amazon DocumentDB entwickeln (mit MongoDB-Kompatibilität), sollten Sie Verbindungen mit Ihrem -Cluster als Replikatsatz herstellen und Lesevorgänge mithilfe der integrierten Leseeinstellungen Ihres Treibers an Replikat-Instances verteilen. In diesem Abschnitt wird eingehender beschrieben, was das bedeutet und wie Sie beispielsweise eine Verbindung mit

Ihrem Amazon DocumentDB DocumentDB-Cluster als Replikatsatz unter Verwendung des SDK für Python herstellen können.

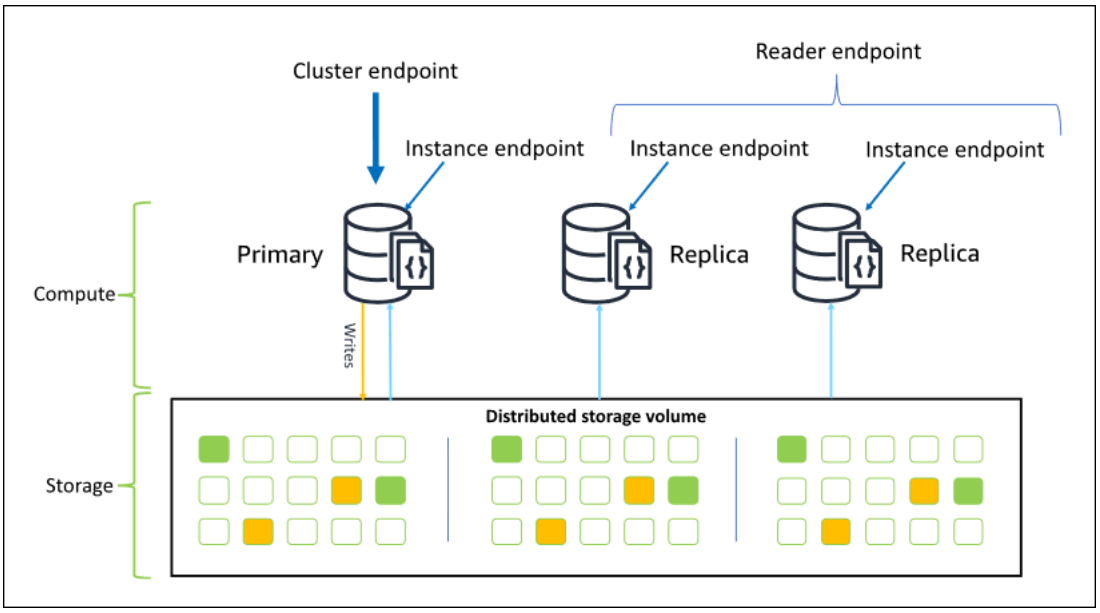
Amazon DocumentDB verfügt über drei Endpunkte, die Sie zum Herstellen von Verbindungen mit Ihrem Cluster verwenden können:

- Cluster-Endpunkt
- Leser-Endpunkt
- Instance-Endpunkte

In den meisten Fällen sollten Sie beim Herstellen von Verbindungen mit Amazon DocumentDB verwenden Sie den Cluster-Endpunkt verwenden. Dies ist ein CNAME, der auf die primäre Instance in Ihrem Cluster verweist wie im folgenden Diagramm gezeigt.

Wenn Sie einen SSH-Tunnel verwenden, empfehlen wir, dass Sie über den Clusterendpunkt eine Verbindung mit Ihrem Cluster herstellen und nicht versuchen, eine Verbindung im Replikatsatzmodus herzustellen (d. h. `replicaSet=rs0` in der Verbindungszeichenfolge anzugeben), da dies zu einem Fehler führt.

> **Note**
>
> Weitere Informationen zu Amazon DocumentDB DocumentDB-Endpunkten finden Sie unterAmazon DocumentDB-Endpunkteaus.



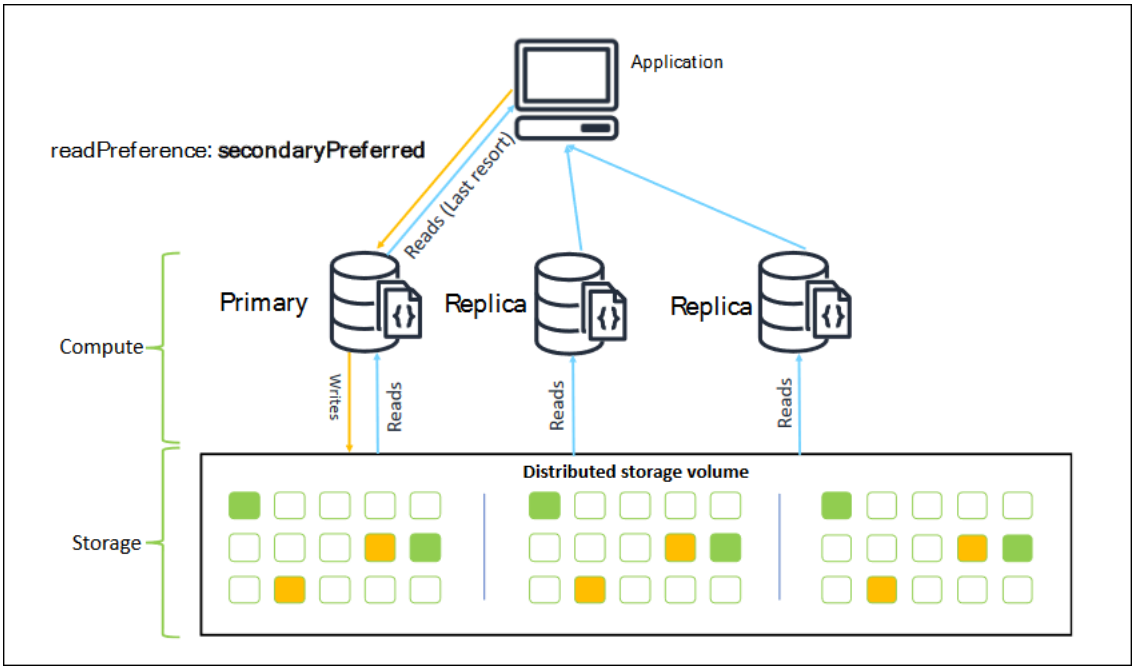
Mithilfe des Cluster-Endpunkts können Sie im Replikatsatzmodus eine Verbindung zu Ihrem Cluster herstellen. Anschließend können Sie die integrierten Treiberfunktionen für Leseeinstellungen verwenden. Im folgenden Beispiel signalisiert die Angabe von `/?replicaSet=rs0` dem SDK, das Sie eine Verbindung als Replikatsatz herstellen möchten. Wenn Sie `/?replicaSet=rs0'` auslassen, leitet der Client alle Anfragen an den Cluster-Endpunkt weiter, d. h. Ihre primäre Instance.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ## replica set and specify the read preference as secondary preferred
   client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0')
```

Der Vorteil der Verbindung als Replikatsatz besteht darin, dass Ihr SDK die Cluster-Topographie automatisch erkennt, auch wenn Instances dem Cluster hinzugefügt oder aus dem Cluster entfernt werden. Anschließend können Sie den Cluster effizienter nutzen, indem Sie Leseanforderungen an Ihre Replikat-Instances weiterleiten.

Wenn Sie eine Verbindung als Replikatsatz herstellen, können Sie die `readPreference` für die Verbindung angeben. Wenn Sie die Leseeinstellung `secondaryPreferred` angeben, leitet der Client Leseabfragen an Ihre Replikate und Schreibabfragen an Ihre primäre Instance weiter (wie im folgenden Diagramm gezeigt). Damit werden Ihre Cluster-Ressourcen besser genutzt. Weitere Informationen finden Sie unter Leseeinstellungsoptionen.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0&readPreference=secondaryPreferred')
```



Lesevorgänge über Amazon DocumentDB DocumentDB-Replikate sind Eventorial. Sie geben die Daten in der gleichen Reihenfolge zurück, in der sie auf dem primären Knoten geschrieben wurden. Die Replikationsverzögerung beträgt häufig weniger als 50 ms. Sie können die Replikationsverzögerung für Ihren Cluster mithilfe der Amazon CloudWatch CloudWatch-Metriken überwachen`DBInstanceReplicaLag`und`DBClusterReplicaLagMaximum`aus. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.

Im Gegensatz zur herkömmlichen monolithischen Datenbankarchitektur trennt Amazon DocumentDB Speicher und Datenverarbeitung. Aufgrund dieser modernen Architektur sollten Sie Lesevorgänge auf Replikat-Instances skalieren. Lesevorgänge auf Replikat-Instances blockieren keine Schreibvorgänge, die von der primären Instance repliziert werden. Sie können einem Cluster bis zu 15 Read-Reaplica-Instances hinzufügen und auf Millionen von Lesevorgängen pro Sekunde skalieren.

Der Hauptvorteil der Verbindung als Replikatsatz und der Verteilung von Lesevorgängen an Replikate besteht darin, dass dies die Gesamtzahl der Ressourcen im Cluster erhöht, die für Ihre Anwendung verfügbar sind. Wir empfehlen die Verbindung als Replikatsatz als bewährte Methode. Darüber hinaus empfehlen wir diesen Ansatz besonders für die folgenden Szenarien:

- Sie nutzen nahezu 100 Prozent der CPU-Leistung auf Ihrer primären Instance.
- Das Puffer-Cache-Treffer-Verhältnis ist beinahe Null.
- Sie erreichen die Verbindungs- oder Cursor-Limits für eine einzelne Instance.

Die Aufwärtsskalierung einer Cluster-Instance-Größe ist eine Option und kann in einigen Fällen die beste Möglichkeit für die Skalierung des Clusters darstellen. Sie sollten jedoch auch überlegen, wie Sie die Replikate, die in Ihrem Cluster bereits vorhanden sind, besser nutzen können. So können Sie die Skalierung erhöhen, ohne dass für Sie höhere Kosten für die Verwendung eines größeren Instance-Typs anfallen. Sie sollten diese Limits darüber hinaus überwachen und Warnungen für sie ausgeben (d. h.`CPUUtilization`,`DatabaseConnections`, und`BufferCacheHitRatio`) Verwenden Sie CloudWatch-Alarme, damit Sie wissen, wann eine Ressource intensiv genutzt wird.

Weitere Informationen finden Sie unter den folgenden Themen:

- Bewährte Methoden für Amazon DocumentDB
- Kontingente und Limits für Amazon DocumentDB

Verwenden von Cluster-Verbindungen

Betrachten Sie ein Szenario, in dem alle Verbindungen in Ihrem Cluster genutzt werden. Beispielsweise hat eine `r5.2xlarge`-Instance ein Limit von 4.500 Verbindungen (und 450 offenen Cursors). Wenn Sie einen Amazon DocumentDB DocumentDB-Cluster mit drei Instances erstellen und nur Verbindungen mit der primären Instance über den Cluster-Endpunkt herstellen, betragen die Cluster-Limits für offene Verbindungen und Cursors 4.500 bzw. 450. Sie können diese Limits erreichen, wenn Sie Anwendungen mit zahlreichen Workern verwenden, die in Containern gestartet werden. Die Container öffnen mehrere Verbindungen gleichzeitig und sättigen den Cluster.

Stattdessen könnten Sie eine Verbindung mit dem Amazon DocumentDB DocumentDB-Cluster als Replikatsatz herstellen und Ihre Lesevorgänge an die Replikat-Instances verteilen. Anschließend könnten Sie die Anzahl der verfügbaren Verbindungen und Cursors im Cluster effektiv auf 13.500 bzw. 1.350 verdreifachen. Das Hinzufügen weiterer Instances zum Cluster erhöht lediglich die Anzahl der Verbindungen und Cursors für Lese-Workloads. Wenn Sie die Anzahl der Verbindungen für Schreibvorgänge in Ihrem Cluster erhöhen müssen, sollten Sie die Instance-Größe erhöhen.

> **Note**
>
> Die Anzahl der Verbindungen für `large`, `xlarge`, und `2xlarge` Instances erhöht sich mit der Instance-Größe bis zu 4.500. Die maximale Anzahl von Verbindungen pro Instance für `4xlarge` Instances oder größer beträgt 4.500. Weitere Hinweise zu Beschränkungen nach Instance-Typ finden Sie unter Instance-Limits.

In der Regel empfehlen wir das Herstellen von Verbindungen mit Ihrem Cluster unter Verwendung der Leseeinstellung `secondary` nicht. Der Grund hierfür ist, dass die Lesevorgänge fehlschlagen, wenn es keine Replikat-Instances im Cluster gibt. Angenommen, Sie führen einen Amazon DocumentDB DocumentDB-Cluster mit zwei Instances aus, einer primären und einer Replikat-Instance. Wenn die Replikat-Instance ein Problem aufweist, schlagen Leseanforderungen aus einem als `secondary` festgelegten Verbindungspool fehl. Der Vorteil von `secondaryPreferred` besteht darin, dass der Client für Lesevorgänge auf den primären Knoten zurückgreift, wenn er keine geeignete Replikat-Instance für die Verbindung finden kann.

Mehrere Verbindungspools

In einigen Szenarien müssen Lesevorgänge in einer Anwendung „Read for Write Consistency“ aufweisen, die nur von der primären Instance in Amazon DocumentDB bereitgestellt werden kann. In diesen Szenarien könnten Sie zwei Client-Verbindungspools erstellen: einen für Schreibvorgänge und einen für Lesevorgänge, die „Read after Write Consistency“ benötigen. Ihr Code würde in diesem Fall ungefähr wie folgt aussehen.

```
## Create a MongoDB client,
##    open a connection to Amazon DocumentDB as a replica set and specify the
 readPreference as primary
clientPrimary = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=primary')

## Create a MongoDB client,
##    open a connection to Amazon DocumentDB as a replica set and specify the
 readPreference as secondaryPreferred
secondaryPreferred = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred')
```

Eine weitere Möglichkeit besteht darin, einen einzelnen Verbindungspool zu erstellen und die Leseeinstellung für eine bestimmte Sammlung zu überschreiben.

```
##Specify the collection and set the read preference level for that collection
col = db.review.with_options(read_preference=ReadPreference.SECONDARY_PREFERRED)
```

Übersicht

Um die Ressourcen im Cluster besser zu nutzen, sollten Sie Verbindungen mit dem Cluster über den Replikatsatzmodus herstellen. Sie können die Zahl der Lesevorgänge für Ihre Anwendung skalieren, indem Sie Ihre Lesevorgänge an die Replikat-Instances verteilen, wenn dies für Ihre Anwendung geeignet ist.

Verbindung zu einem Amazon DocumentDB-Cluster von außerhalb einer Amazon VPC herstellen

Amazon DocumentDB-Cluster (mit MongoDB-Kompatibilität) werden innerhalb einer Amazon Virtual Private Cloud Umgebung Cloud VirtualVirtualVirtualVirtualVirtualVirtualVirtualVirtualVirtualVirtualVirtualVirtual Auf sie kann direkt von Amazon EC2 EC2-Instances oder anderenAWS Services zugegriffen werden, die in derselben Amazon VPC bereitgestellt werden. Darüber hinaus kann über EC2-Instances oder andereAWS Services in verschiedenen VPCs in derselbenAWS-Region oder anderen Regionen über VPC-Peering auf Amazon DocumentDB zugegriffen werden.

Nehmen wir jedoch an, dass Ihr Anwendungsfall erfordert, dass Sie (oder Ihre Anwendung) von außerhalb der VPC des Clusters auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zugreifen. In diesem Fall können Sie SSH-Tunneling (auch Portweiterleitung genannt) verwenden, um auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen.

Die eingehende Diskussion zum Thema SSH-Tunneling geht über den Rahmen dieses Themas hinaus. Weitere Informationen zum SSH-Tunneling finden Sie in den folgenden Ressourcen:

- SSH-Tunnel
- Beispiel zum SSH-Port-Forwarding, insbesondere der Abschnitt Lokales Forwarding.

Um einen SSH-Tunnel zu erstellen, benötigen Sie eine Amazon EC2 Instance in derselben Amazon VPC ausgeführt wird wie Ihr Amazon DocumentDB-Cluster ausgeführt wird. Sie können entweder eine vorhandene EC2-Instance in derselben VPC wie Ihr Cluster verwenden oder eine erstellen. Weitere Informationen finden Sie im entsprechenden Thema für Ihr Betriebssystem:

- Erste Schritte mit Amazon EC2 Linux-Instances
- Erste Schritte mit Amazon EC2 Windows-Instances

Normalerweise verwenden Sie vermutlich den folgenden Befehl, um sich mit einer EC2-Instance zu verbinden:

```
ssh -i "ec2Access.pem" ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com
```

In diesem Fall können Sie einen SSH-Tunnel zum Amazon DocumentDB-Cluster einrichten,`sample-cluster.node.us-east-1.docdb.amazonaws.com` indem Sie den folgenden Befehl auf Ihrem lokalen Computer ausführen. Das `-L`-Flag dient zur Weiterleitung eines lokalen Ports. Wenn Sie einen SSH-Tunnel verwenden, empfehlen wir, dass Sie über den Clusterendpunkt eine Verbindung mit Ihrem Cluster herstellen und nicht versuchen, eine Verbindung im Replikatsatzmodus herzustellen (d. h. `replicaSet=rs0` in der Verbindungszeichenfolge anzugeben), da dies zu einem Fehler führt.

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-east-1.docdb.amazonaws.com:27017
 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -N
```

Nachdem der SSH-Tunnel erstellt wurde, werden alle Befehle, die Sie ausgeben, an den Amazon DocumentDB-Cluster weitergeleitet, der in der Amazon VPC`sample-cluster` ausgeführt wird.`localhost:27017` Wenn Transport Layer Security (TLS) in Ihrem Amazon DocumentDB-Cluster aktiviert ist, müssen Sie den öffentlichen Schlüssel für Amazon DocumentDB von herunterladenhttps://truststore.pki.rds.amazonaws.com/global/global-bundle.pem. Der folgende Vorgang lädt diese Datei herunter:

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

> **Note**
>
> TLS ist standardmäßig für neue Amazon DocumentDB-Cluster aktiviert. Eine Deaktivierung ist jedoch möglich. Weitere Informationen finden Sie unter TLS-Einstellungen für Amazon DocumentDB-Cluster verwalten.

Verwenden Sie den folgenden Befehl, um von außerhalb der Amazon VPC aus eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen.

```
mongo --sslAllowInvalidHostnames --ssl --sslCAFile global-bundle.pem --username
 <yourUsername> --password <yourPassword>
```

Von Studio 3T aus eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen

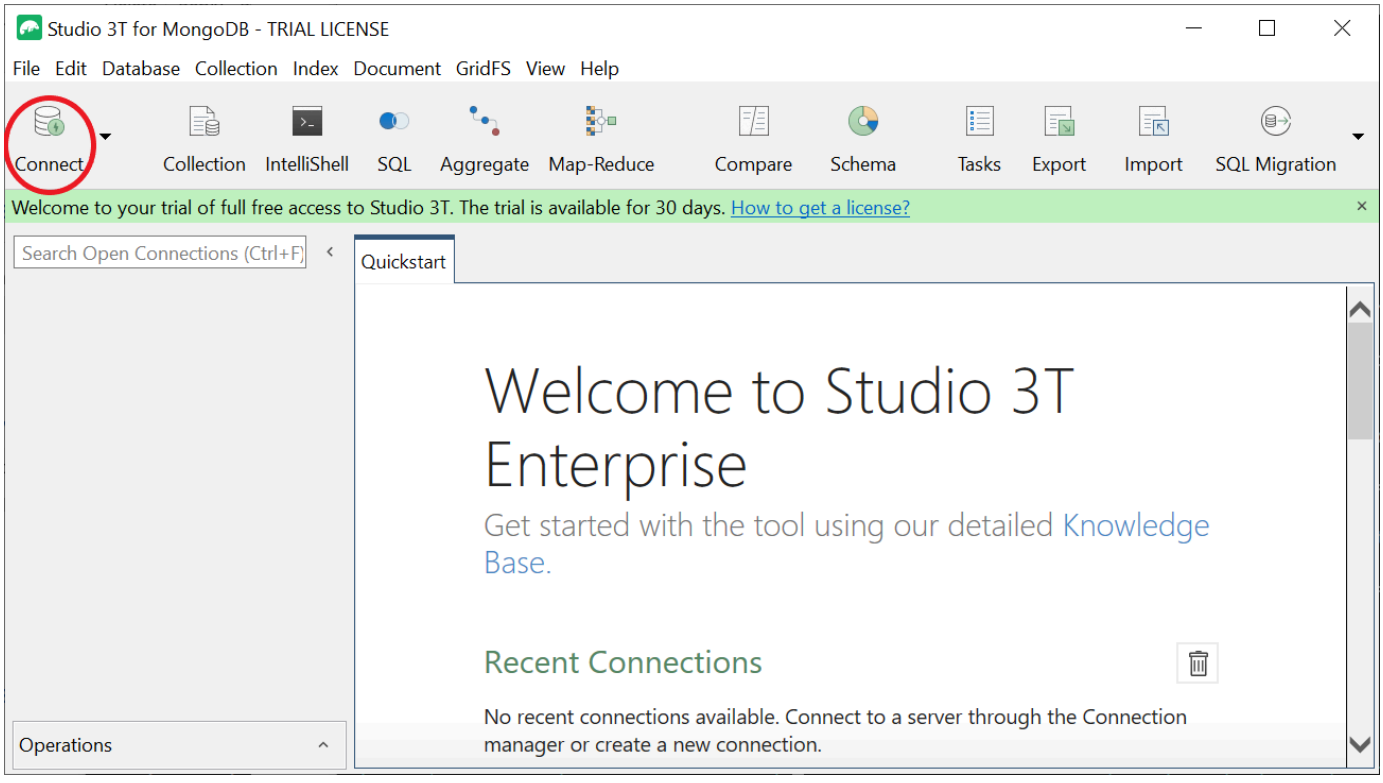
Studio 3T ist eine beliebte GUI und IDE für Entwickler und Dateningenieure, die mit MongoDB arbeiten. Es bietet mehrere leistungsstarke Funktionen Baum-, Tabellen- und JSON-Ansichten Ihrer Daten, einfachen Import/Export in CSV, JSON, SQL und BSON/MongoDump, flexible Abfrageoptionen, eine visuelle drag-and-drop Benutzeroberfläche, eine integrierte Mongo-Shell mit Autovervollständigung, einen Aggregationspipeline-Editor und SQL-Abfrageunterstützung.

Voraussetzungen

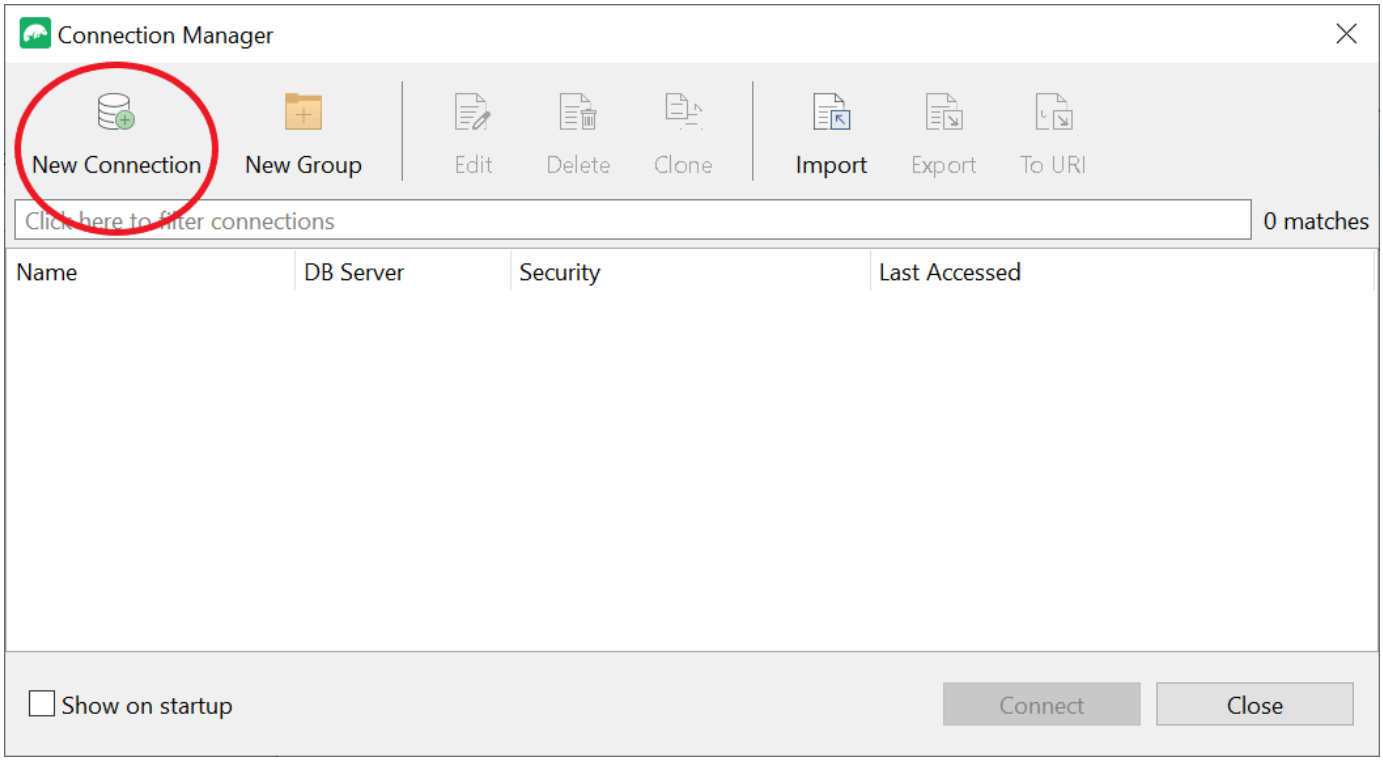
- Wenn auf Amazon EC2 noch kein Amazon DocumentDB-Cluster läuft, folgen Sie den Anweisungen zum Herstellen einer Verbindung mit Amazon EC2.
- Wenn Sie Studio 3T nicht haben, laden Sie es herunter und installieren Sie es.

Mit Studio 3T Connect

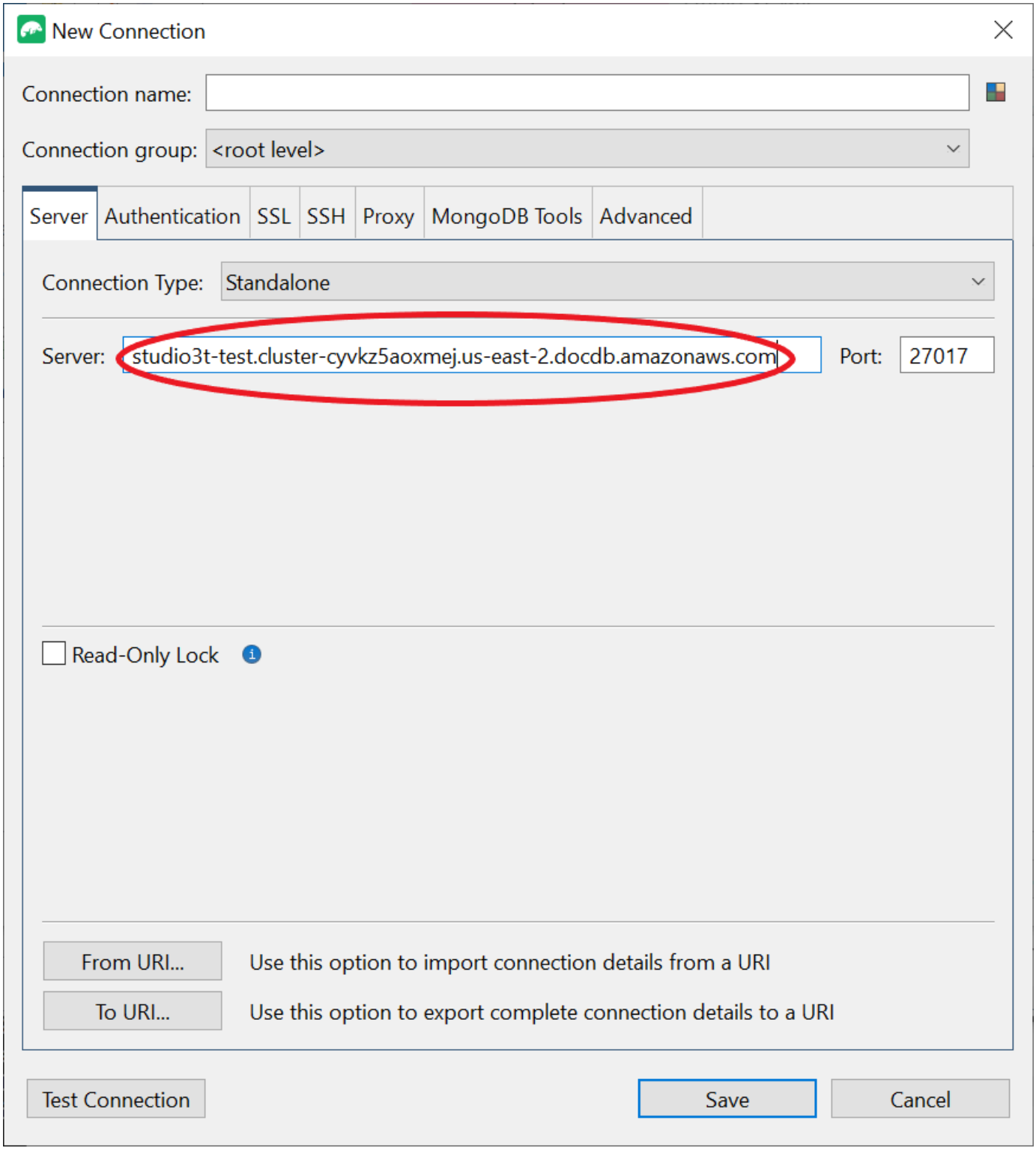
1. Wählen Sie in der oberen linken Ecke der Werkzeugleiste Connect.



2. Wählen Sie in der oberen linken Ecke der Werkzeugleiste „Neue Verbindung“.

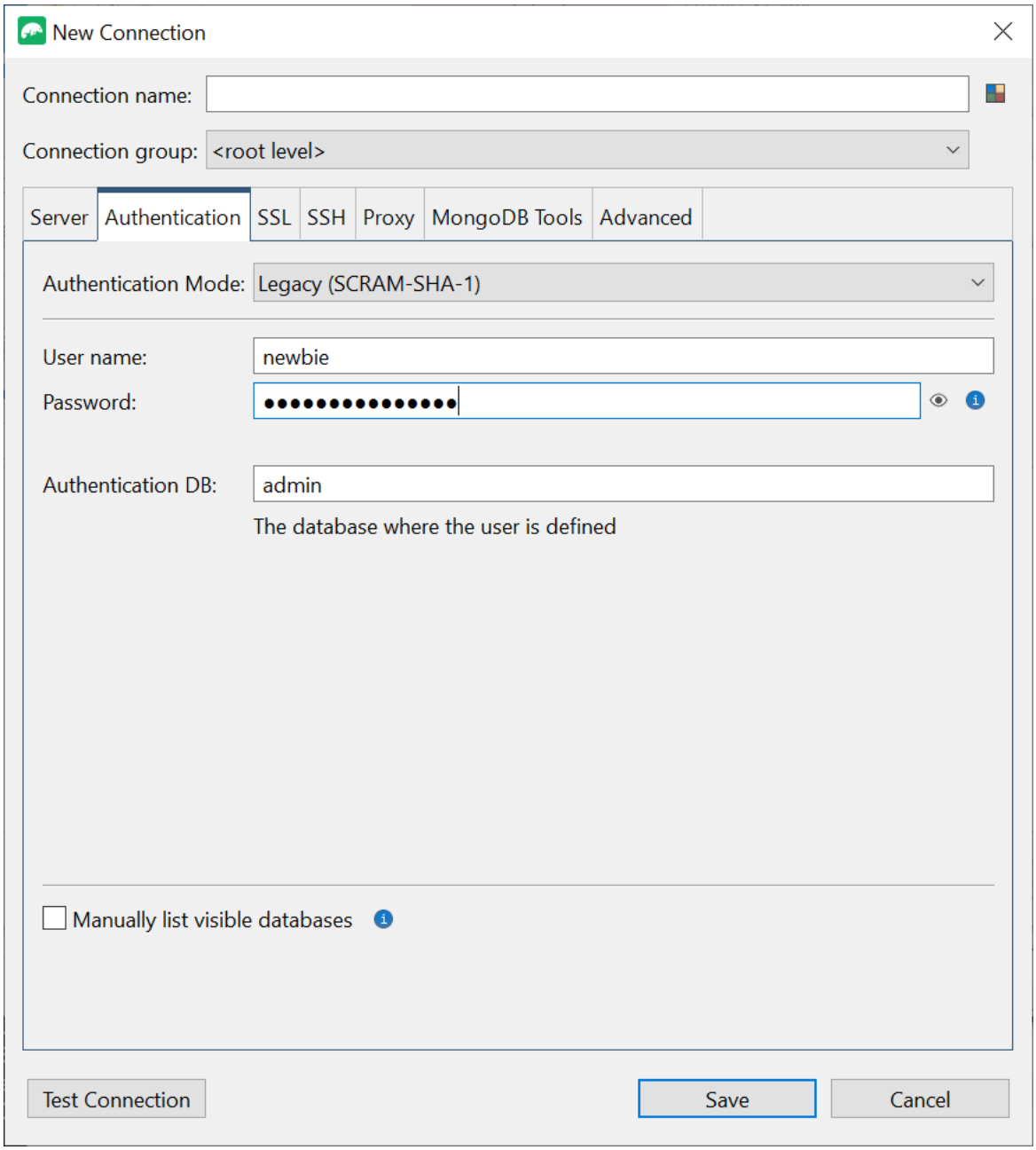


3. Geben Sie auf der Registerkarte Server im Feld Server die Informationen zum Cluster-Endpunkt ein.



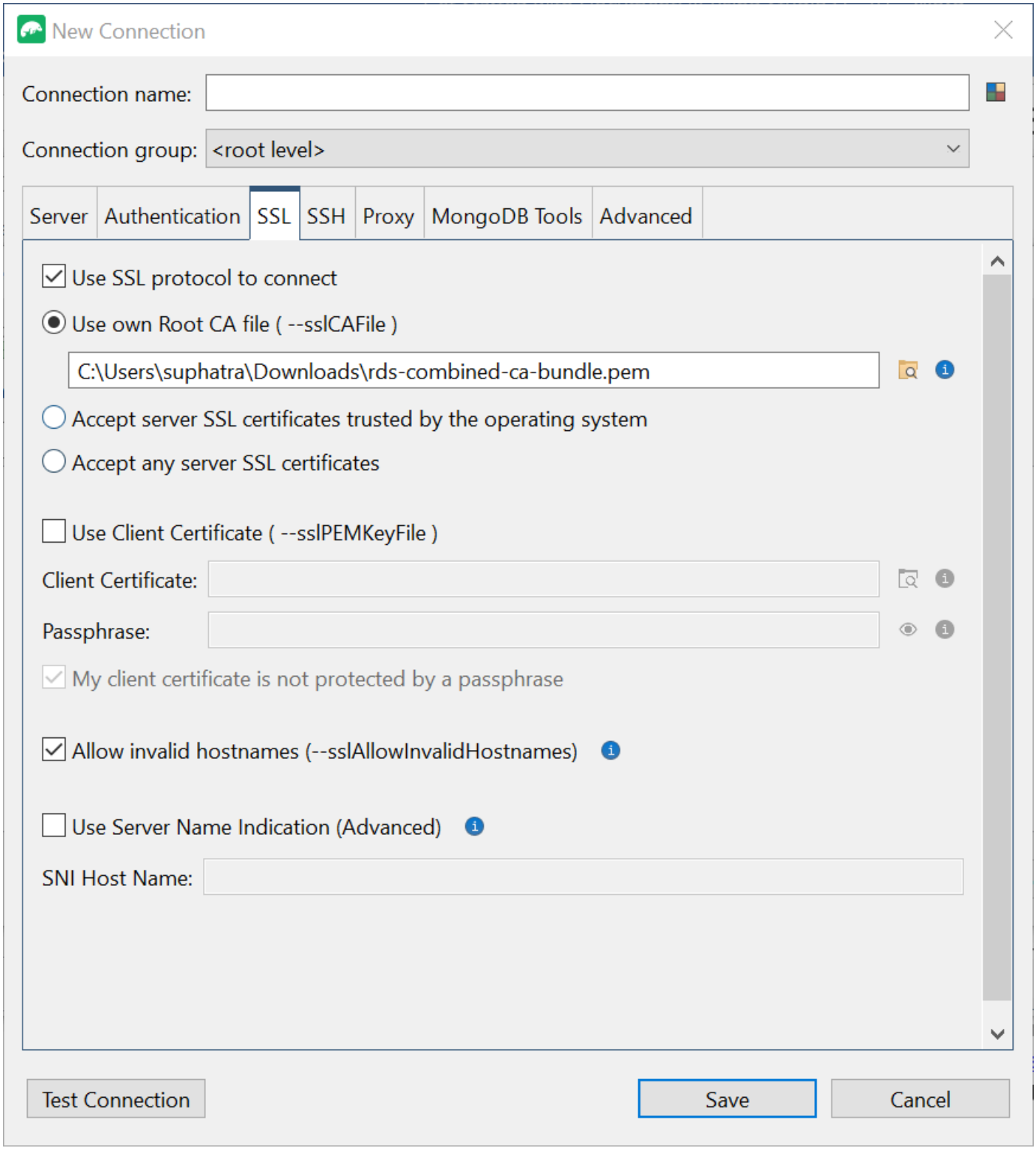
> **Note**
>
> Sie können Ihren Cluster-Endpunkt nicht finden? Folgen Sie einfach den Schritten hier.

4. Wählen Sie die Registerkarte Authentifizierung und wählen Sie Legacy im Drop-down-Menü für Authentifizierungsmodus aus.

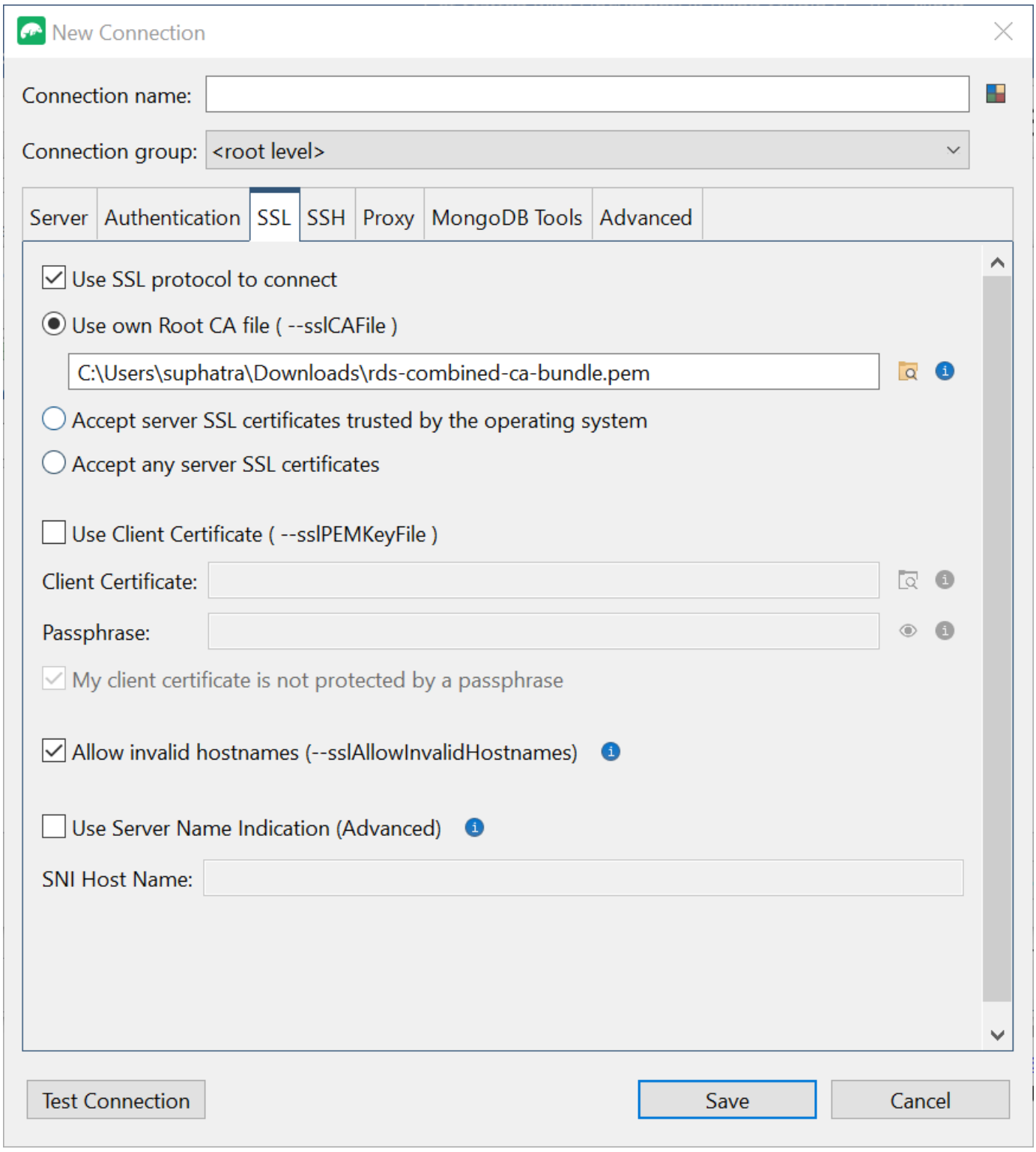


5. Geben Sie Ihren Benutzernamen und Ihre Anmeldeinformationen in die Felder Benutzername und Passwort ein.

6. Wählen Sie die Registerkarte SSL und aktivieren Sie das Kästchen SSL-Protokoll für die Verbindung verwenden.



7. Wählen Sie Eigene Root-CA-Datei verwenden. Fügen Sie dann das Amazon DocumentDB DocumentDB-Zertifikat hinzu (Sie können diesen Schritt überspringen, wenn SSL in Ihrem DocumentDB-Cluster deaktiviert ist). Markieren Sie das Kästchen, um ungültige Hostnamen zuzulassen.



> **Note**
>
> Sie haben das Zertifikat nicht? Sie können es mit dem folgenden Befehl herunterladen:

```
wget https://truststore.pki.rds.amazonaws.com/global/global-
bundle.pem
```

8. Wenn Sie von einem Client-Computer außerhalb der Amazon VPC aus eine Verbindung herstellen, müssen Sie einen SSH-Tunnel erstellen. Sie werden dies auf der Registerkarte SSH tun.

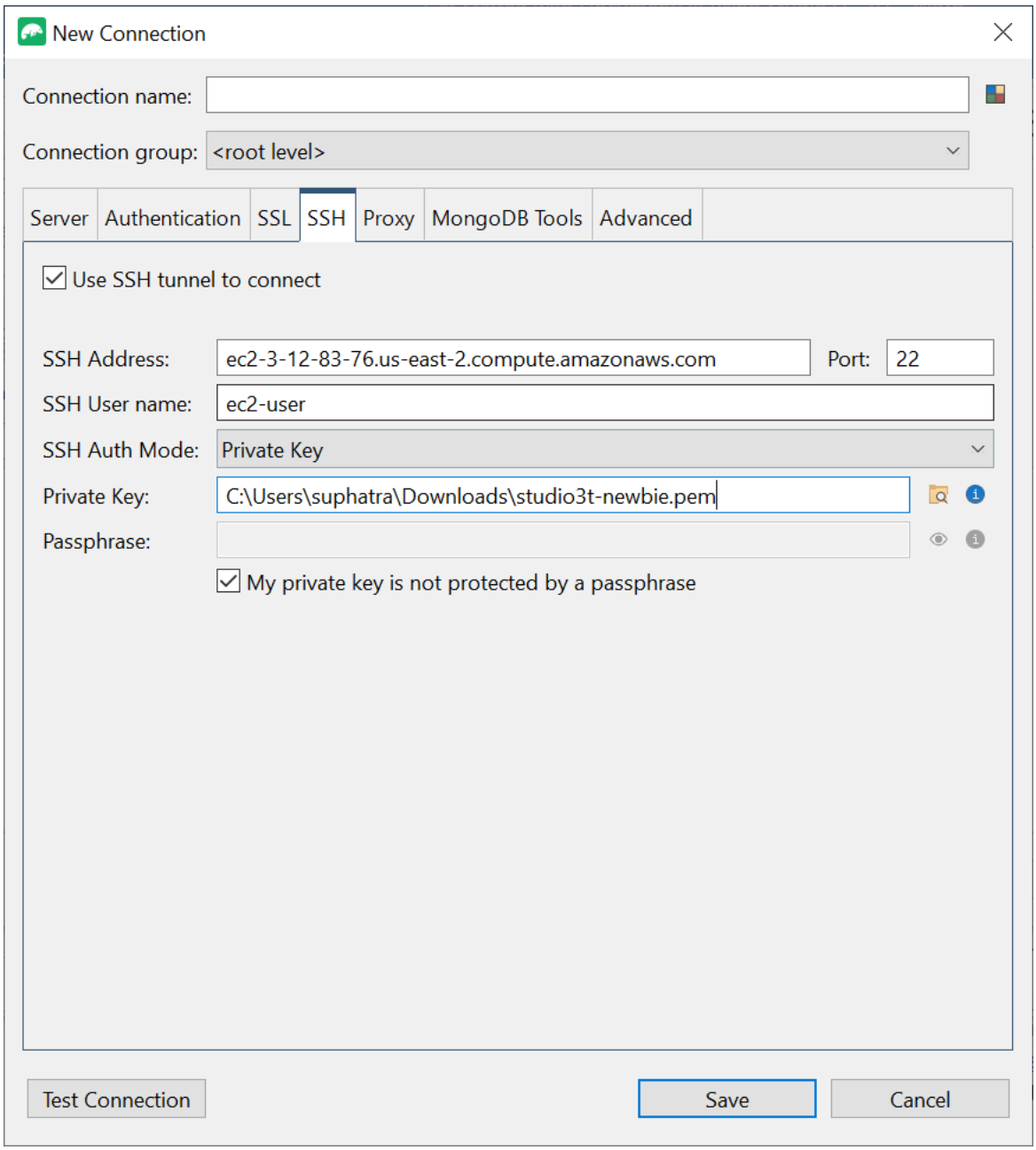
   a. Aktivieren Sie das Kontrollkästchen SSH-Tunnel verwenden und geben Sie die SSH-Adresse in das Feld SSH-Adresse ein. Dies ist Ihre Instanz Public DNS (IPV4). Sie können diese URL von Ihrer Amazon EC2-Managementkonsole abrufen.

   b. Geben Sie Ihren Nutzernamen ein. Dies ist der Benutzername Ihrer Amazon EC2 EC2-Instance

   c. Wählen Sie für den SSH-Authentifizierungsmodus die Option Private Key aus. Wählen Sie im Feld Privater Schlüssel das Dateifinder-Symbol aus, um den privaten Schlüssel Ihrer Amazon EC2 EC2-Instance zu suchen und auszuwählen. Dies ist die PEM-Datei (key pair), die Sie beim Erstellen Ihrer Instance in der Amazon EC2 EC2-Konsole gespeichert haben.

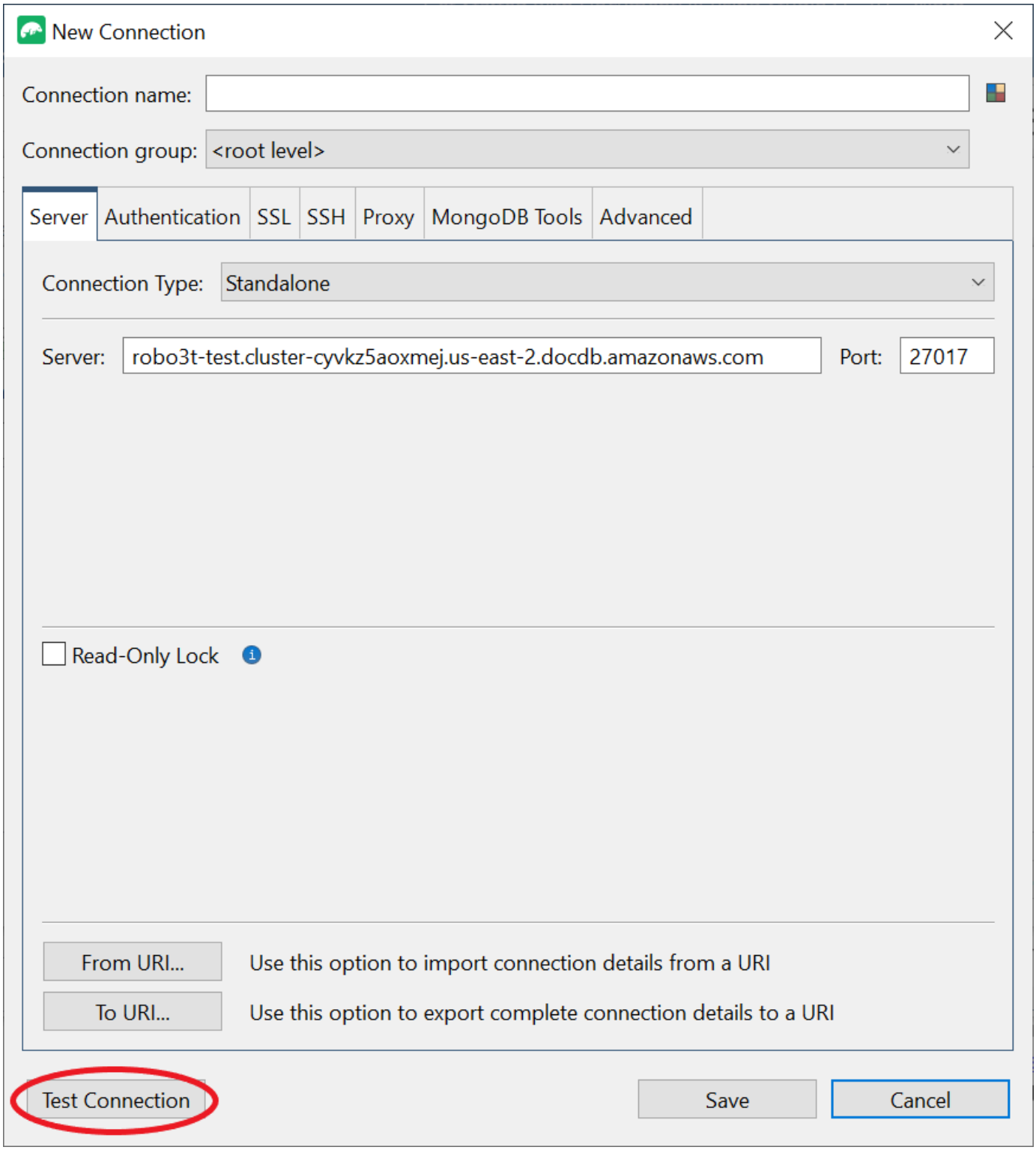
   d. Wenn Sie sich auf einem Linux/macOS-Client-Computer befinden, müssen Sie möglicherweise die Berechtigungen Ihres privaten Schlüssels mit dem folgenden Befehl ändern:

      ```
      chmod 400 /fullPathToYourPemFile/<yourKey>.pem
      ```

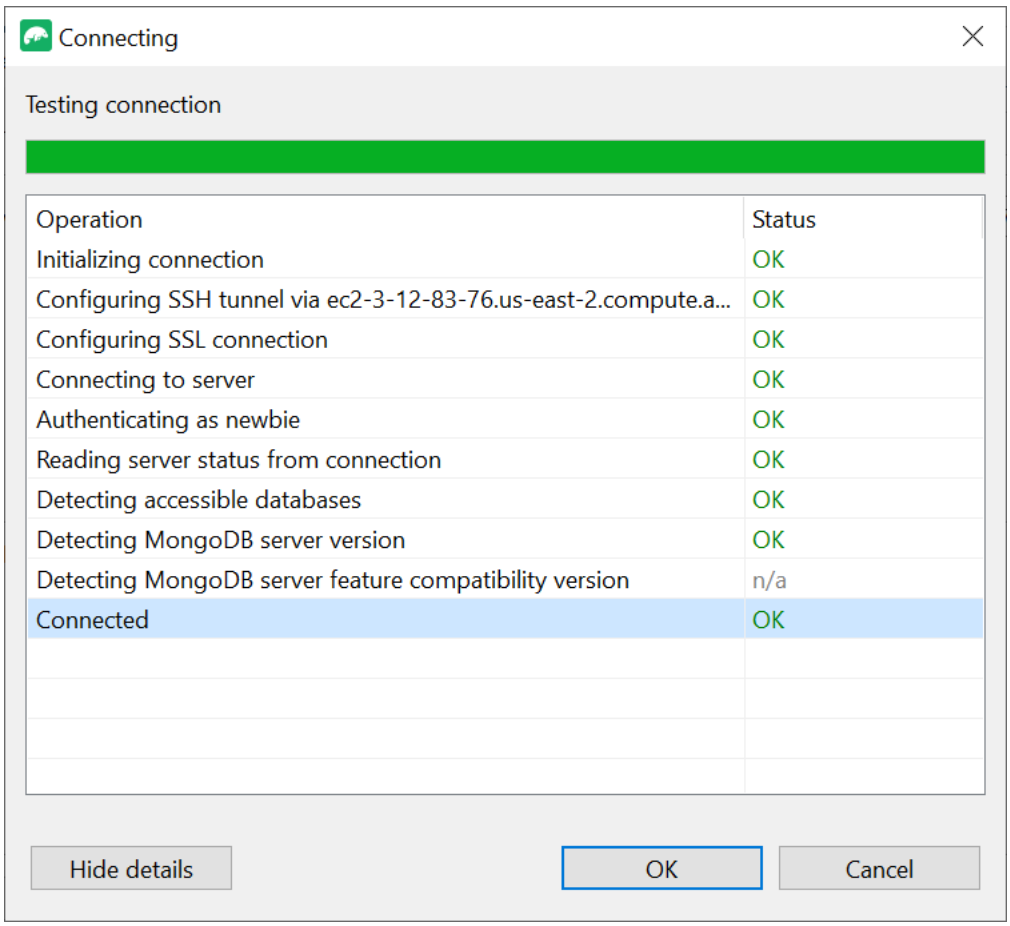


> **Note**
>
> Diese Amazon EC2 EC2-Instance sollte sich in derselben Amazon VPC und Sicherheitsgruppe wie Ihr DocumentDB-Cluster befinden. Sie können die SSH-Adresse, den Benutzernamen und den privaten Schlüssel von Ihrer Amazon EC2-Managementkonsole abrufen.

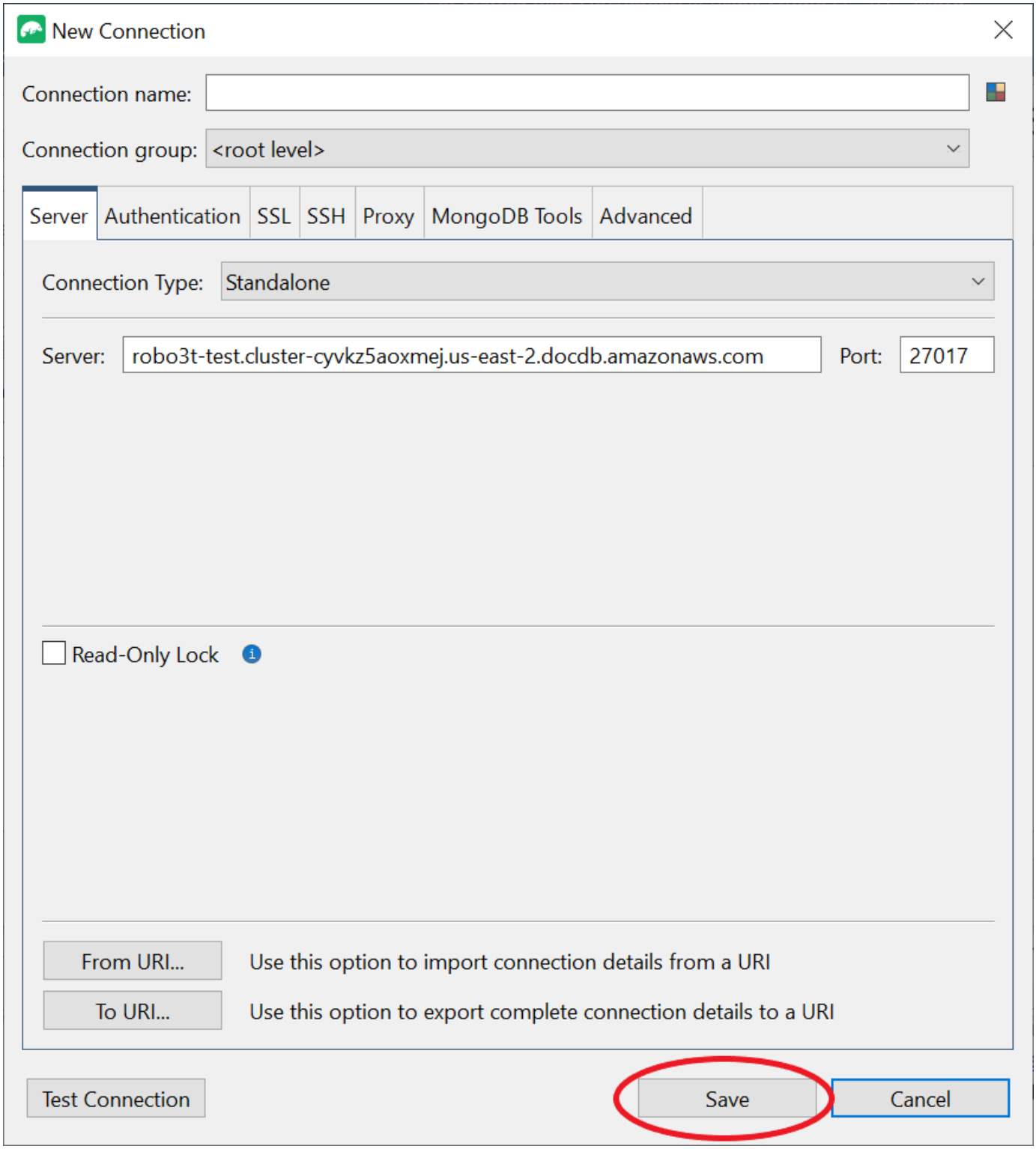
9. Testen Sie nun Ihre Konfiguration, indem Sie auf die Schaltfläche Verbindung testen klicken.



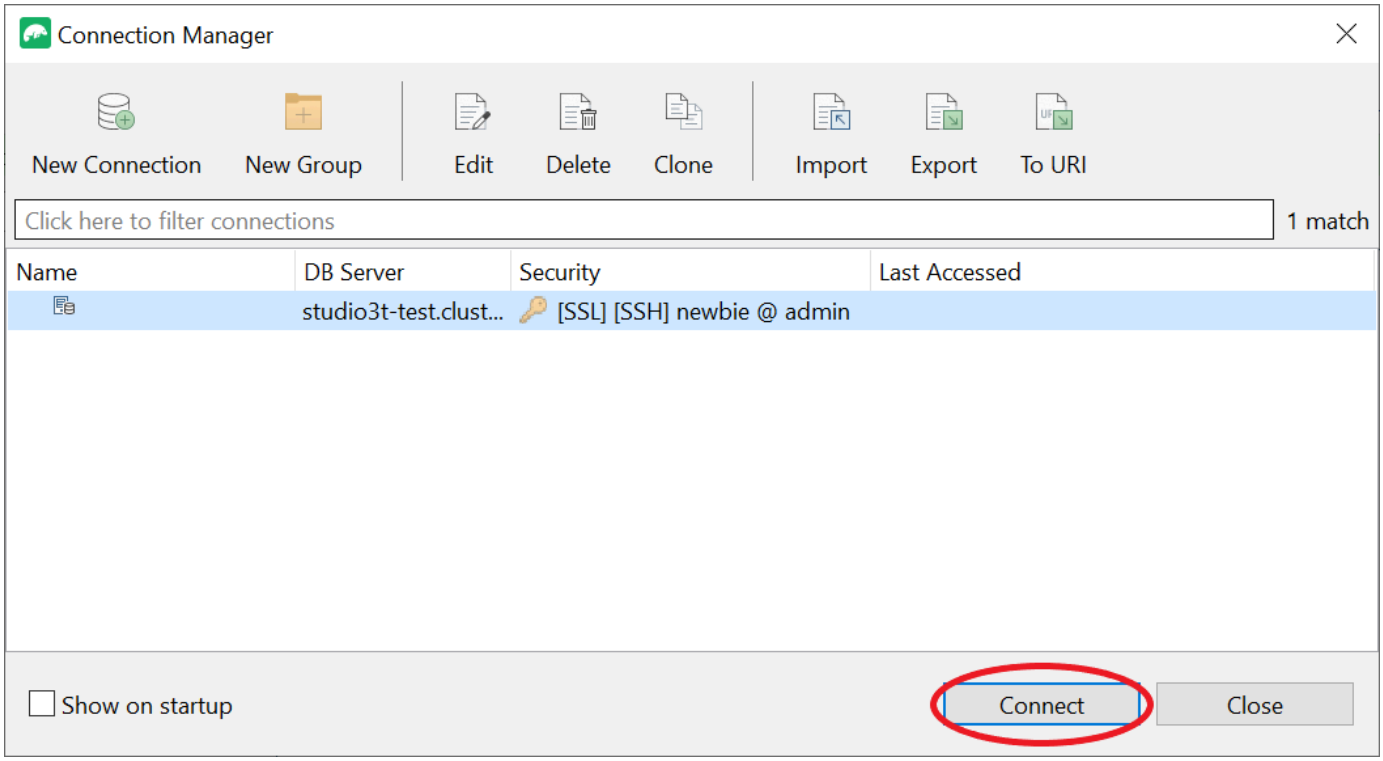
10. In einem Diagnosefenster sollte ein grüner Balken angezeigt werden, der anzeigt, dass der Test erfolgreich war. Wählen Sie nun OK, um das Diagnosefenster zu schließen.



11. Wählen Sie Speichern, um Ihre Verbindung für die future Verwendung zu speichern.



12. Wählen Sie nun Ihren Cluster aus und wählen Sie Connect.



Herzlichen Glückwunsch! Sie sind jetzt erfolgreich über Studio 3T mit Ihrem Amazon DocumentDB-Cluster verbunden.

Connect zu Amazon DocumentDB her mit DataGrip

DataGripist eine leistungsstarke integrierte Entwicklungsumgebung (IDE), die verschiedene Datenbanksysteme unterstützt, darunter Amazon DocumentDB. Dieser Abschnitt führt Sie durch die Schritte, mit denen Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen DataGrip, sodass Sie Ihre Daten einfach über eine grafische Oberfläche verwalten und abfragen können.

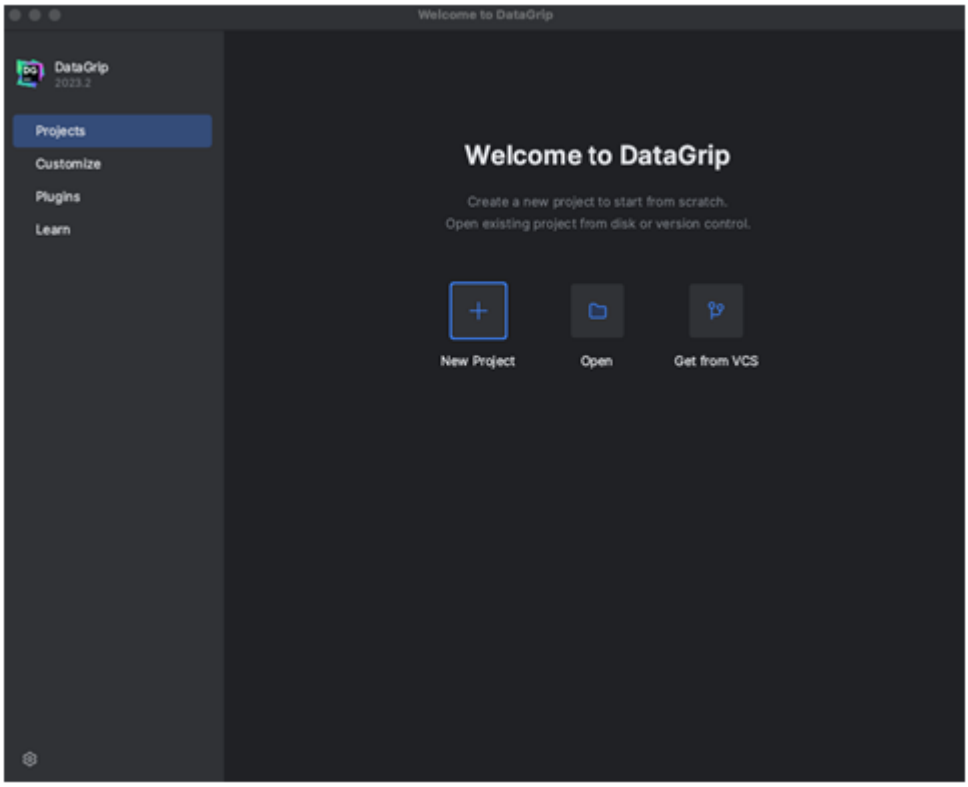
Voraussetzungen

- DataGrip IDE ist auf Ihrem Computer installiert. Sie können es von herunterladen JetBrains.
- Eine Amazon EC2 EC2-Instance, die in derselben VPC wie Ihr Amazon DocumentDB-Cluster läuft. Sie verwenden diese Instance, um einen sicheren Tunnel von Ihrem lokalen Computer zum Amazon DocumentDBCluster einzurichten. Folgen Sie den Anweisungen dazu. Herstellen einer Verbindung über Amazon EC2

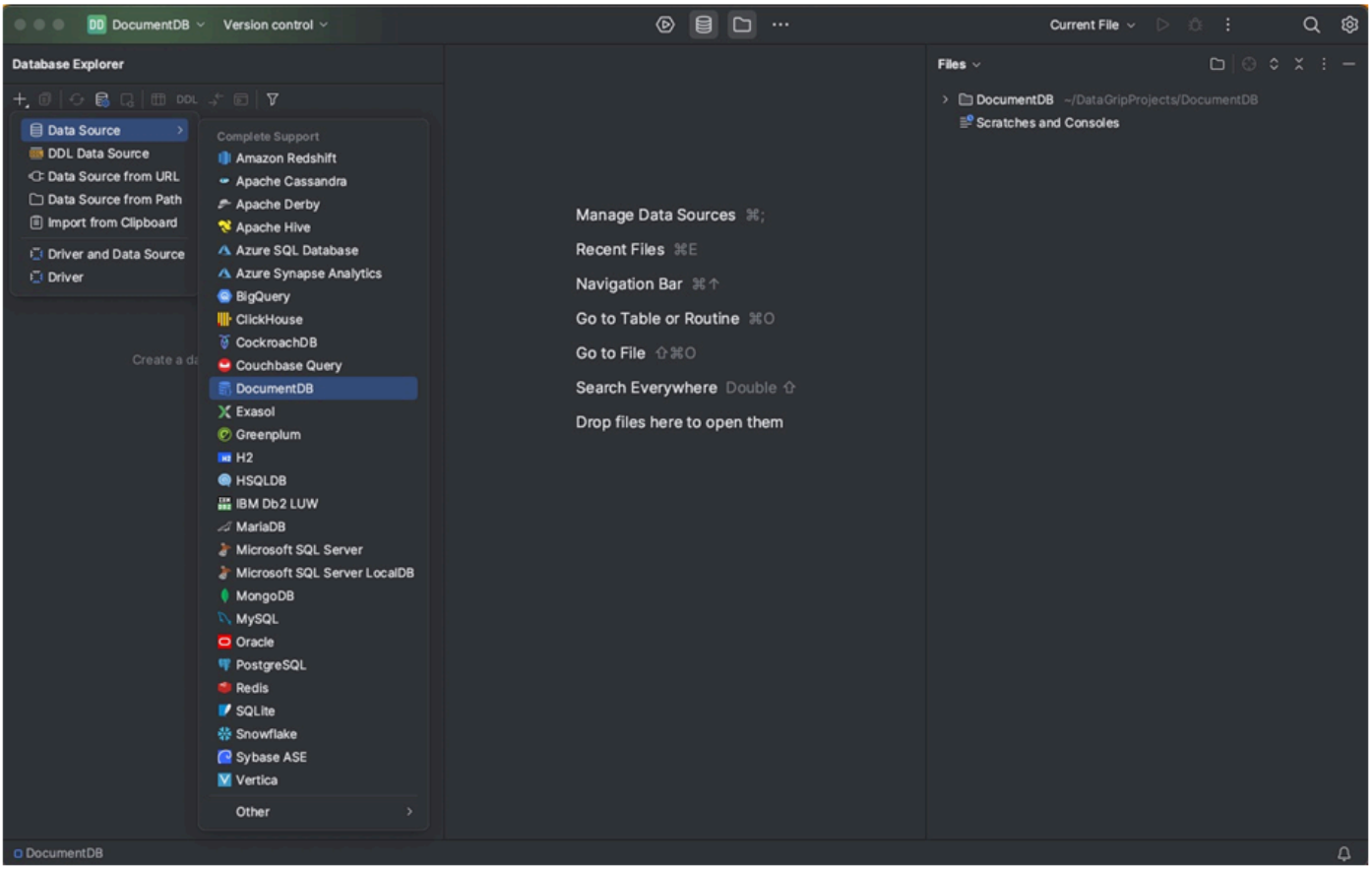
- Alternative zu einer Amazon EC2 EC2-Instance, einer VPN-Verbindung oder wenn Sie bereits über ein sicheres VPN auf Ihre AWS Infrastruktur zugreifen. Wenn Sie diese Option bevorzugen, folgen Sie den Anweisungen für den sicheren Zugriff auf Amazon DocumentDB mit AWS Client VPN.

Connect mit DataGrip

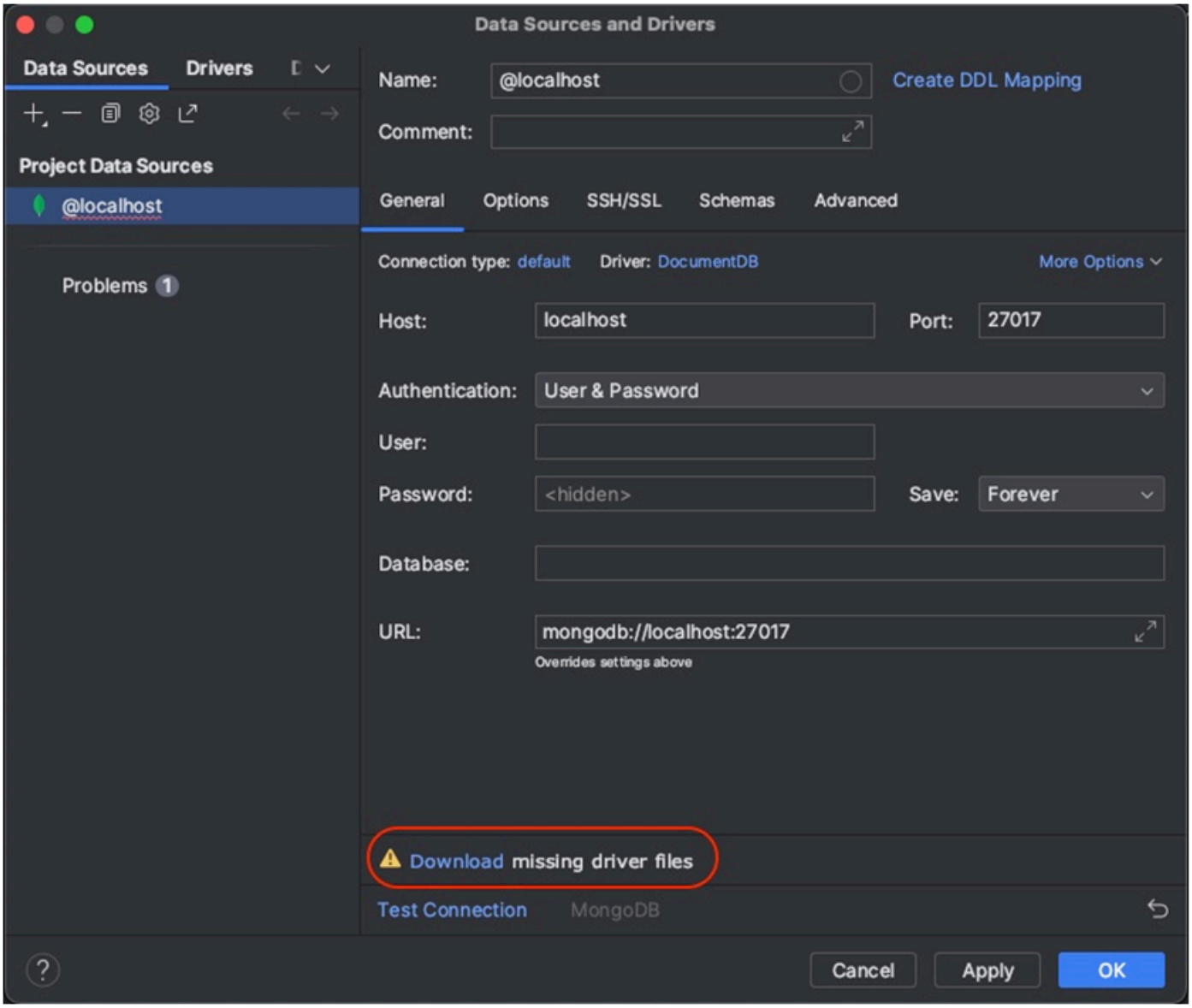
1. Starten Sie DataGrip auf Ihrem Computer und erstellen Sie ein neues Projekt.



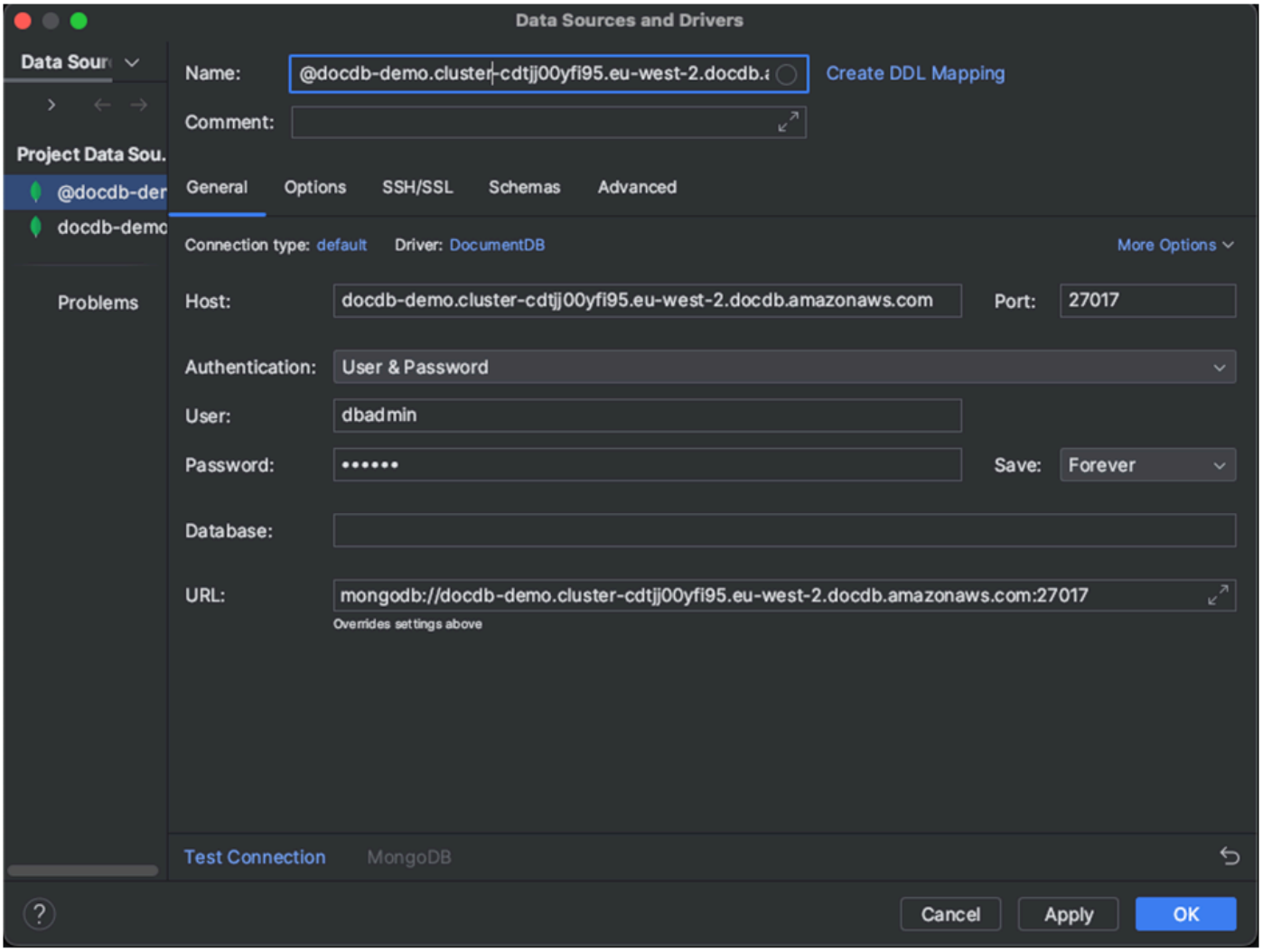
2. Fügen Sie mithilfe einer der folgenden Methoden eine neue Datenquelle hinzu:
   a. Navigieren Sie im Hauptmenü zu Datei — Neu — Datenquelle und wählen Sie DocumentDB
   b. Klicken Sie im Datenbank-Explorer in der Werkzeugleiste auf das neue Symbol (+). Navigieren Sie zu Data Source und wählen Sie DocumentDB aus.



3. Überprüfen Sie auf der Seite Datenquellen auf der Registerkarte Allgemein, ob sich unten im Bereich mit den Verbindungseinstellungen der Link Fehlende Treiberdateien herunterladen befindet. Klicken Sie auf diesen Link, um Treiber herunterzuladen, die für die Interaktion mit einer Datenbank erforderlich sind. Einen direkten Download-Link finden Sie unter JetBrains JDBC-Treiber.



4. Geben Sie auf der Registerkarte Allgemein die Verbindungsdetails an:
   a. Geben Sie im Feld Host den Amazon DocumentDB-Cluster-Endpunkt an.
   b. Der Port ist bereits auf 27017 eingestellt. Ändern Sie es, wenn Ihr Cluster auf einem anderen Port bereitgestellt wurde.
   c. Wählen Sie für Authentifizierung die Option Benutzer und Passwort aus.
   d. Geben Sie Ihren Benutzernamen und Ihr Passwort ein.
   e. Das Datenbankfeld ist optional. Sie können die Datenbank angeben, zu der Sie eine Verbindung herstellen möchten.
   f. Das URL-Feld wird automatisch vervollständigt, wenn Sie die obigen Details hinzufügen.



5. Aktivieren Sie auf der Registerkarte SSH/SSL die Option SSH-Tunnel verwenden und klicken Sie dann auf das Symbol, um den SSH-Konfigurationsdialog zu öffnen. Geben Sie die folgenden Informationen ein:

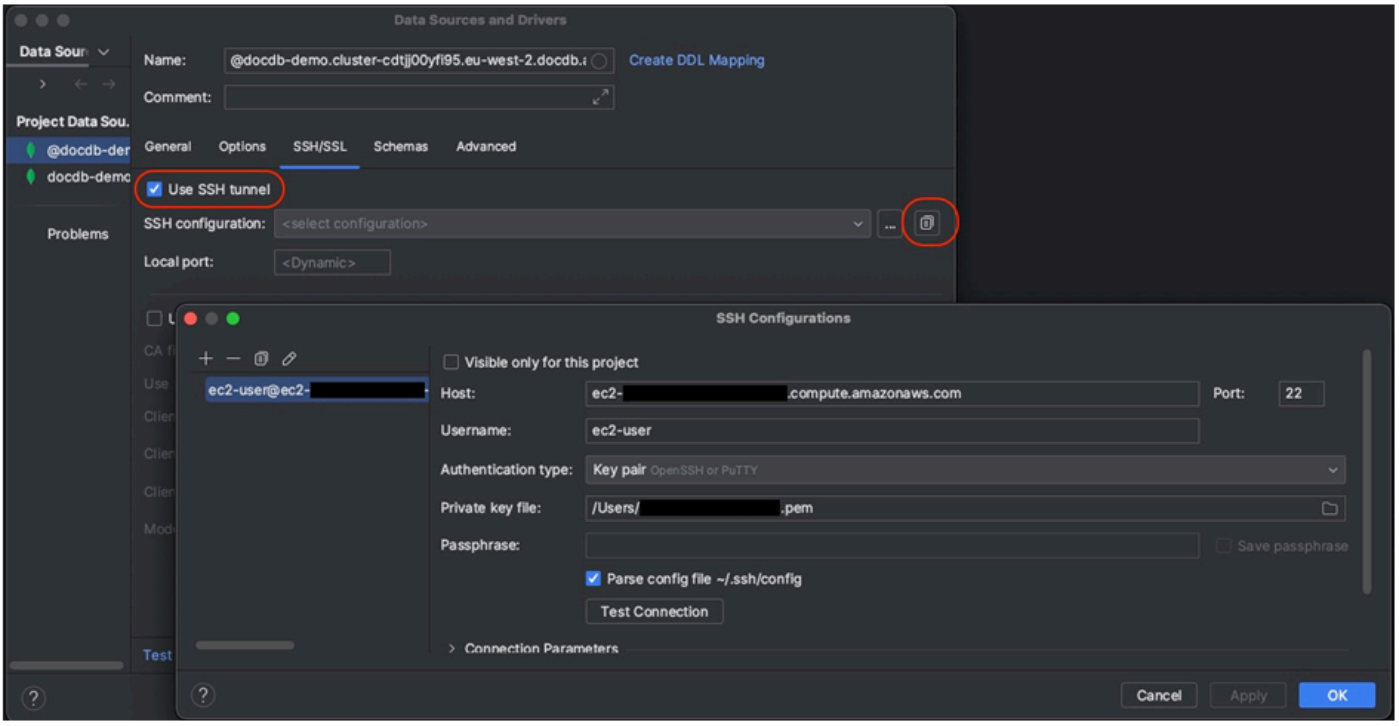
   a. geben Sie im Feld Host den Hostnamen Ihrer Amazon EC2 EC2-Instance ein.

   b. Geben Sie den Benutzernamen und das Passwort für Ihre Amazon EC2 EC2-Instance ein.

   c. Wählen Sie als Authentifizierungstyp die Option Schlüsselpaar aus.

   d. Geben Sie Ihre private Schlüsseldatei ein.

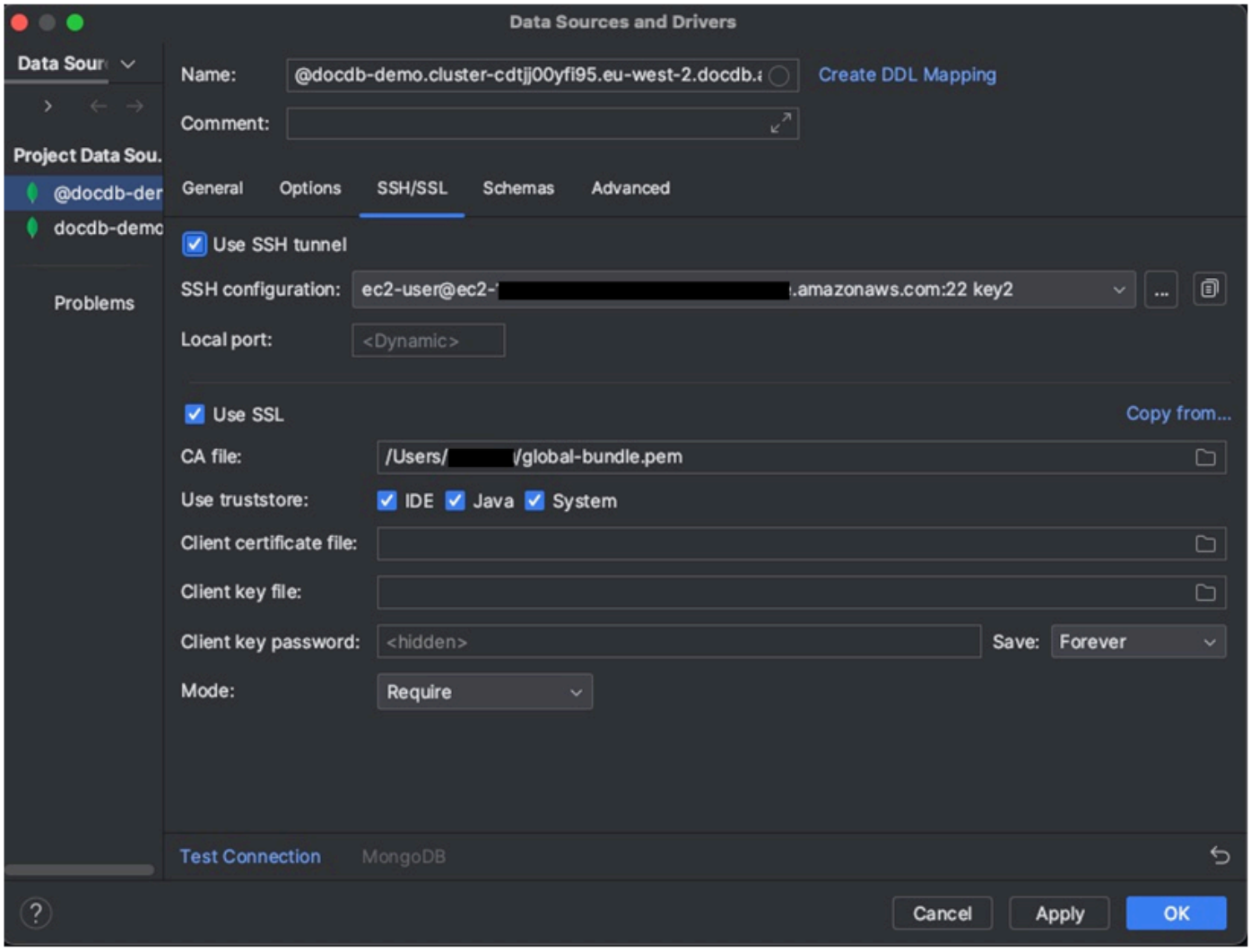
> **Note**
>
> Wenn Sie die VPN-Option verwenden, müssen Sie den SSH-Tunnel nicht konfigurieren.



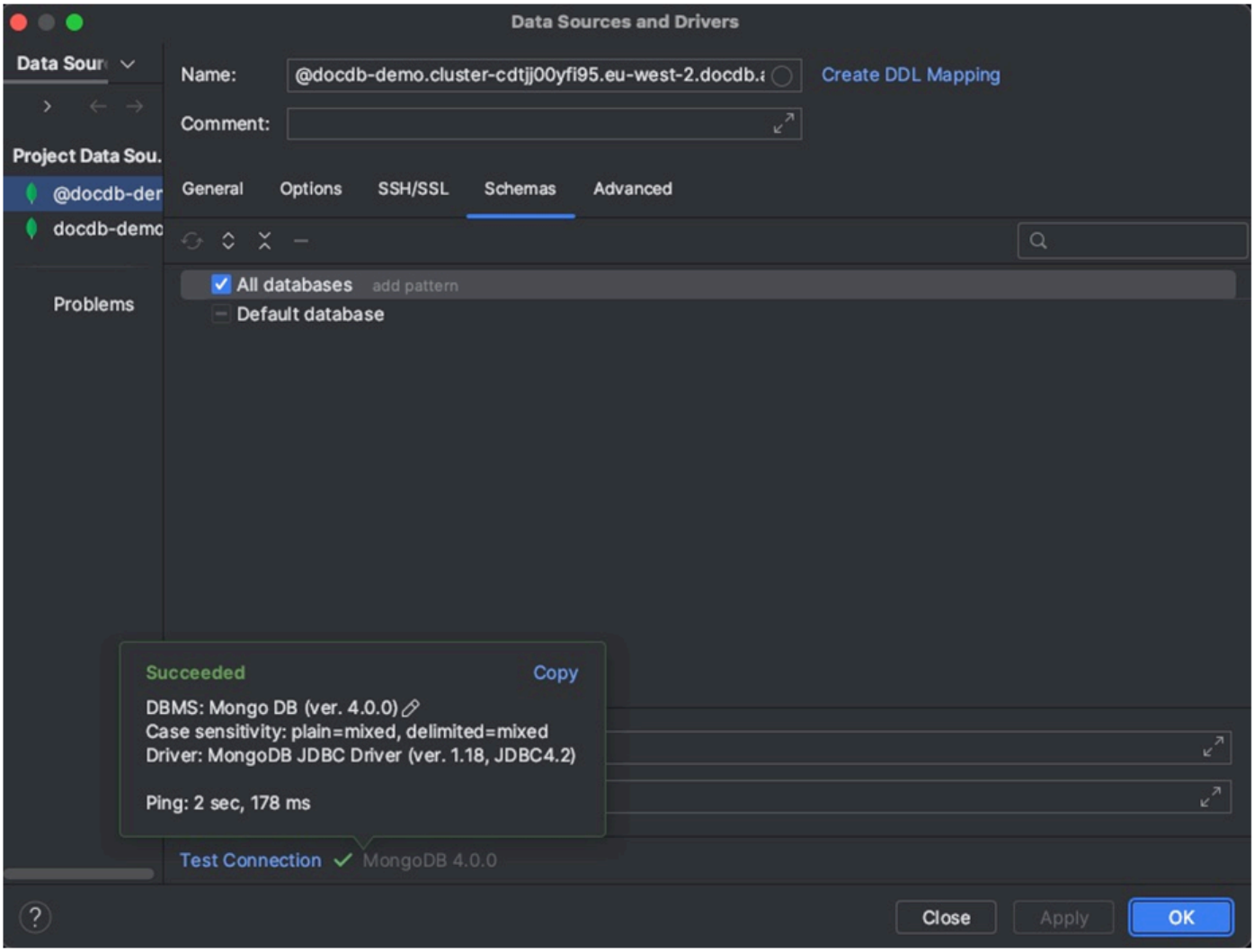
6. Aktivieren Sie auf der Registerkarte SSH/SSL die Option SSL verwenden. Geben Sie im Feld CA-Datei den Speicherort der `global-bundle.pem` Datei auf Ihrem Computer ein. Behalten Sie für Modus die Option Erforderlich bei.

> **Note**
>
> Sie können das Zertifikat von diesem Speicherort oder mit dem folgenden Befehl herunterladen: wget https://aws.amazon.com/https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem

> **Note**
>
> Wenn Sie eine Verbindung zu Amazon DocumentDB Elastic Cluster herstellen, müssen Sie die CA-Datei nicht angeben. Lassen Sie die Option SSL verwenden aktiviert und lassen Sie alle anderen Optionen auf ihren Standardwerten stehen.



7. Wählen Sie auf der Registerkarte Schemas die Option Alle Datenbanken aus, oder geben Sie den Filter „*: *“ in das Feld Schemamuster ein. Klicken Sie auf den Link Verbindung testen, um die Verbindung zu testen.



8. Sobald die Verbindung erfolgreich getestet wurde, klicken Sie auf OK, um die Datenquellenkonfiguration zu speichern.

DataGrip Funktionen

DataGrip bietet verschiedene Funktionen, die Ihnen helfen, effizient mit Amazon DocumentDB zu arbeiten:

- SQL Editor — Schreiben Sie SQL-ähnliche Abfragen in Ihren DocumentDB-Sammlungen und führen Sie sie mit dem SQL-Editor in aus. DataGrip
- Visual Query Builder — Verwenden Sie den Visual Query Builder, um Abfragen grafisch zu erstellen, ohne SQL-Code schreiben zu müssen.

- Schemaverwaltung — Einfache Verwaltung Ihres Datenbankschemas, einschließlich des Erstellens, Änderns und Löschens von Sammlungen.
- Datenvisualisierung — Zeigen Sie Ihre Daten an und analysieren Sie sie mithilfe verschiedener Visualisierungstools, die unter verfügbar sind. DataGrip
- Daten exportieren und importieren — Übertragen Sie Daten zwischen Amazon DocumentDB und anderen Datenbanken mithilfe DataGrip der Export- und Importfunktionen.

Weiterführende Funktionen und Tipps zur Arbeit mit Amazon DocumentDB und anderen Datenbanksystemen finden Sie in der offiziellen DataGrip Dokumentation.

Herstellen einer Verbindung über Amazon EC2

In diesem Abschnitt wird beschrieben, wie Sie die Konnektivität zwischen einem Amazon DocumentDB-Cluster und Amazon EC2 einrichten und von der Amazon-ECAmazon EC2 aus auf den Amazon DocumentDB-Cluster zugreifen.

Es gibt zwei Möglichkeiten, die EC2-Verbindung zu konfigurieren:

- Automatisches Verbinden Ihrer EC2-Instance mit einer Amazon DocumentDB-Datenbank – Verwenden Sie die automatische Verbindungsfunktion in der EC2-Konsole, um die Verbindung zwischen Ihrer EC2-Instance und einer neuen oder vorhandenen Amazon DocumentDB-Datenbank automatisch zu konfigurieren. Diese Verbindung ermöglicht den Datenverkehr zwischen der EC2-Instance und der Amazon DocumentDB-Datenbank. Diese Option wird normalerweise zum Testen und Erstellen neuer Sicherheitsgruppen verwendet.
- Manuelles Verbinden Ihrer EC2-Instance mit Ihrer Amazon DocumentDB-Datenbank – Konfigurieren Sie die Verbindung zwischen Ihrer EC2-Instance und Ihrer Amazon DocumentDB-Datenbank, indem Sie die Sicherheitsgruppen manuell konfigurieren und zuweisen, um die Konfiguration zu reproduzieren, die durch das automatische Verbindungsfeature erstellt wird. Diese Option wird in der Regel zum Ändern erweiterter Einstellungen und zum Verwenden vorhandener Sicherheitsgruppen verwendet.

Voraussetzungen

Unabhängig von der Option und bevor Sie Ihren ersten Amazon DocumentDB-Cluster erstellen, müssen Sie Folgendes tun:

Erstellen eines Amazon Web Services (AWS)-Kontos

Bevor Sie Amazon DocumentDB verwenden können, benötigen Sie ein Amazon Web Services (AWS)-Konto. Das AWS Konto ist kostenlos. Sie zahlen nur für die Services und Ressourcen, die Sie wirklich nutzen.

Wenn Sie kein haben AWS-Konto, führen Sie die folgenden Schritte aus, um eines zu erstellen.

So registrieren Sie sich für ein AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für ein registrieren AWS-Konto, Root-Benutzer des AWS-Kontos wird ein erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Richten Sie die erforderlichen AWS Identity and Access Management (IAM)-Berechtigungen ein.

Für den Zugriff auf die Verwaltung von Amazon DocumentDB-Ressourcen wie Cluster, Instances und Cluster-Parametergruppen sind Anmeldeinformationen erforderlich, die zur Authentifizierung Ihrer Anforderungen verwenden AWS kann. Weitere Informationen finden Sie unter Identity and Access Management für Amazon DocumentDB.

1. Geben Sie in die Suchleiste von IAM AWS Management Console ein und wählen Sie IAM im daraufhin angezeigten Dropdown-Menü aus.
2. Sobald Sie sich in der IAM-Konsole befinden, wählen Sie im Navigationsbereich Benutzer aus.
3. Wählen Sie Ihren Benutzernamen aus.
4. Klicken Sie auf die Schaltfläche Berechtigungen hinzufügen .
5. Wählen Sie die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.

6. Geben Sie `AmazonDocDBFullAccess` in die Suchleiste ein und wählen Sie sie aus, sobald sie in den Suchergebnissen erscheint.
7. Klicken Sie unten auf die blaue Schaltfläche, auf der Weiter: Überprüfen steht.
8. Klicken Sie unten auf die blaue Schaltfläche mit der Meldung Berechtigungen hinzufügen .

Erstellen einer Amazon Virtual Private Cloud (Amazon VPC)

Je nachdem, in welchem AWS-Region Sie sich befinden, ist möglicherweise bereits eine Standard-VPC erstellt. Wenn Sie keine Standard-VPC haben, führen Sie Schritt 1 von Erste Schritte mit Amazon VPC im Amazon-VPC-Benutzerhandbuch aus. Dies dauert weniger als fünf Minuten.

Automatisches Verbinden von Amazon EC2

Themen

- Automatisches Verbinden einer EC2-Instance mit einer neuen Amazon DocumentDB-Datenbank
- Automatisches Verbinden einer EC2-Instance mit einer vorhandenen Amazon DocumentDB-Datenbank
- Übersicht über die automatische Verbindung mit einer EC2-Instance
- Anzeigen verbundener Rechenressourcen

Bevor Sie eine Verbindung zwischen einer EC2-Instance und einer neuen Amazon DocumentDB-Datenbank einrichten, stellen Sie sicher, dass Sie die unter beschriebenen Anforderungen erfüllenÜbersicht über die automatische Verbindung mit einer EC2-Instance. Wenn Sie nach der Konfiguration der Konnektivität Änderungen an Sicherheitsgruppen vornehmen, können sich die Änderungen auf die Verbindung zwischen der EC2-Instance und der Amazon DocumentDB-Datenbank auswirken.

> **Note**
>
> Sie können eine Verbindung zwischen einer EC2-Instance und einer Amazon DocumentDB-Datenbank nur automatisch einrichten, indem Sie die verwenden AWS Management Console. Sie können eine Verbindung nicht automatisch mit der AWS CLI oder der Amazon DocumentDB-API einrichten.

Automatisches Verbinden einer EC2-Instance mit einer neuen Amazon DocumentDB-Datenbank

Im folgenden Verfahren wird davon ausgegangen, dass Sie die Schritte im Voraussetzungen Thema abgeschlossen haben.

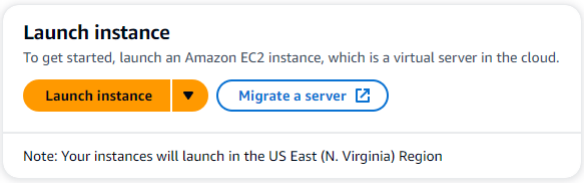
Schritte

- Schritt 1: Erstellen einer Amazon EC2-Instance
- Schritt 2: Erstellen eines Amazon DocumentDB-Clusters
- Schritt 3: Herstellen einer Verbindung mit Ihrer Amazon EC2-Instance
- Schritt 4: Installieren der mongo-Shell
- Schritt 5: Verwalten von Amazon DocumentDB TLS
- Schritt 6: Herstellen einer Verbindung mit Ihrem Amazon DocumentDB-Cluster
- Schritt 7: Einfügen und Abfragen von Daten
- Schritt 8: Erkunden

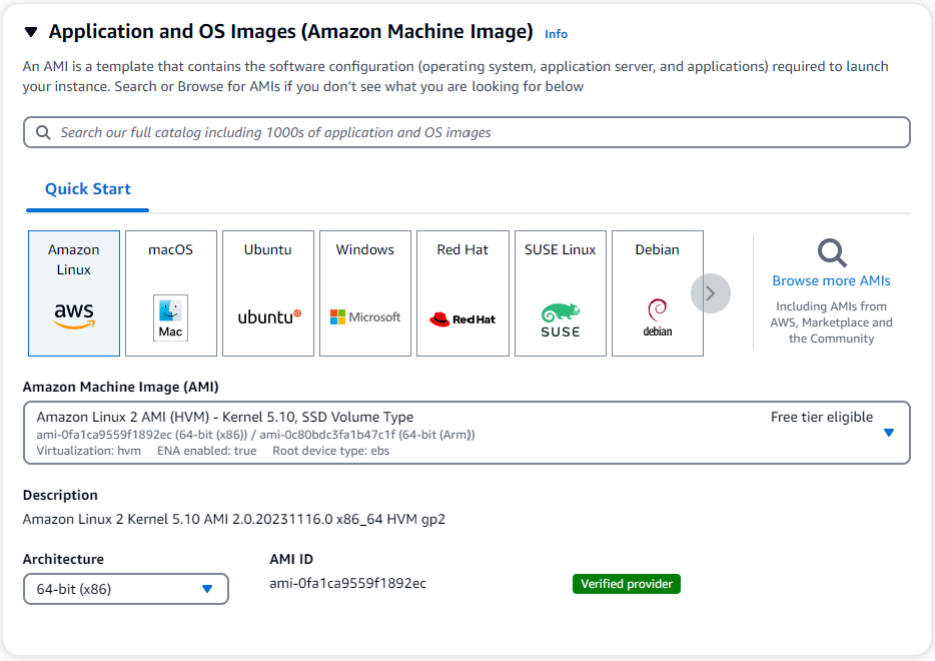
Schritt 1: Erstellen einer Amazon EC2-Instance

In diesem Schritt erstellen Sie eine Amazon EC2-Instance in derselben Region und Amazon VPC, die Sie später zur Bereitstellung Ihres Amazon DocumentDB-Clusters verwenden werden.

1. Wählen Sie in der Amazon EC2-Konsole Instance starten aus.

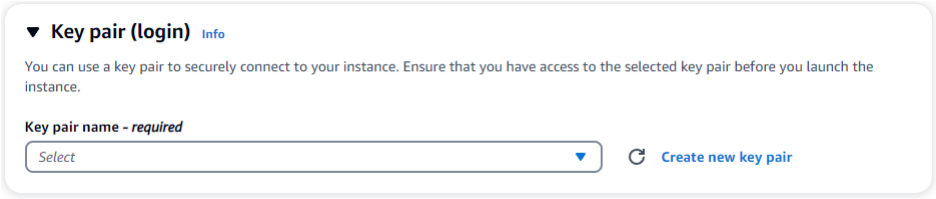


2. Geben Sie einen Namen oder eine Kennung in das Feld Name ein, das sich im Abschnitt Name und Tags befindet.
3. Suchen Sie in der Dropdown-Liste Amazon Machine Image (AMI) nach Amazon Linux 2 AMI und wählen Sie es aus.



4. Suchen Sie t3.micro in der Dropdown-Liste Instance-Typ und wählen Sie es aus.

5. Geben Sie im Abschnitt Schlüsselpaar (Anmeldung) die Kennung eines vorhandenen Schlüsselpaars ein oder wählen Sie Neues Schlüsselpaar erstellen aus.



Sie müssen ein Amazon EC2-Schlüsselpaar angeben.

Wenn Sie über ein Amazon EC2-Schlüsselpaar verfügen:

a. Wählen Sie ein Schlüsselpaar und dann Ihr Schlüsselpaar aus der Liste aus.

b. Sie müssen bereits über die private Schlüsseldatei (PEM- oder PPK-Datei) verfügen, um sich bei Ihrer Amazon EC2-Instance anzumelden.

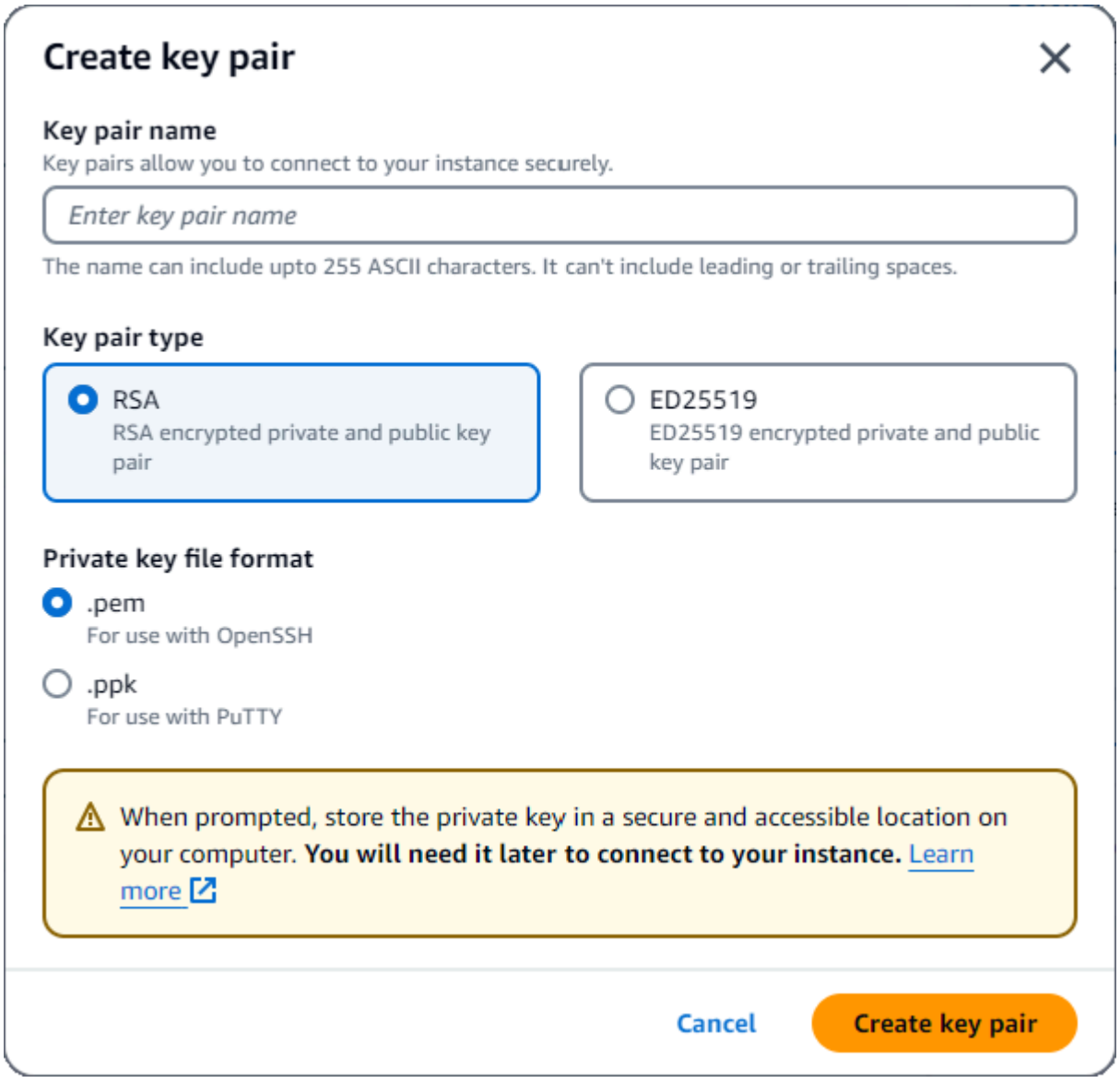
Wenn Sie kein Amazon EC2-Schlüsselpaar haben:

a. Wählen Sie Neues Schlüsselpaar erstellen aus, das Dialogfeld Schlüsselpaar erstellen wird angezeigt.

b. Geben Sie einen Namen in das Feld Schlüsselpaarname ein.

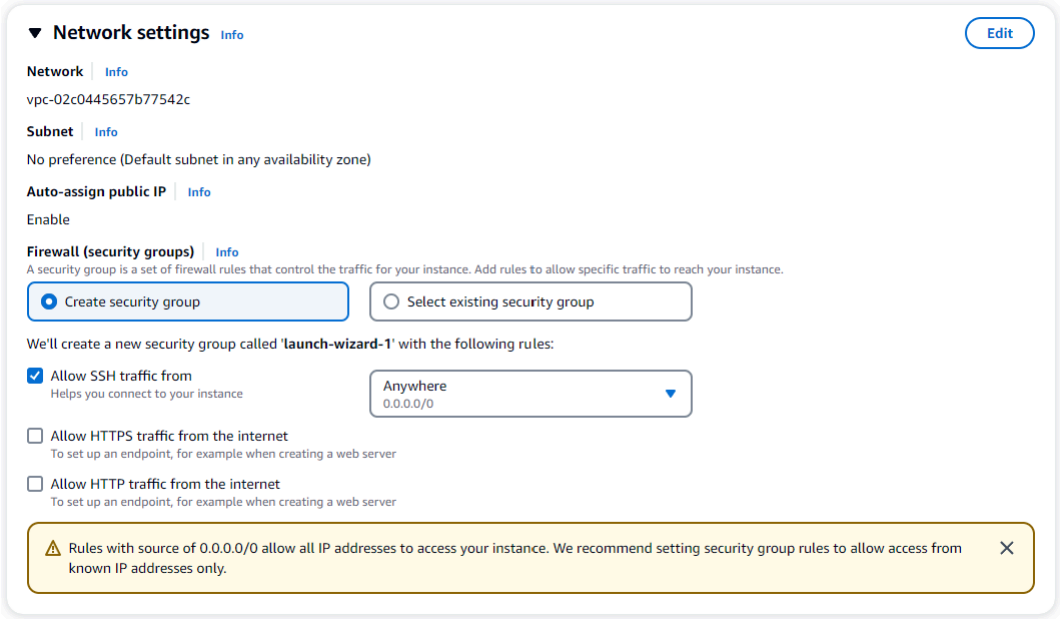
c. Wählen Sie den Schlüsselpaartyp und das Dateiformat für den privaten Schlüssel aus.

d. Wählen Sie Create Key Pair (Schlüsselpaar erstellen) aus.



> **Note**
>
> Aus Sicherheitsgründen empfehlen wir dringend, ein Schlüsselpaar für SSH- und Internetkonnektivität zu Ihrer EC2-Instance zu verwenden.

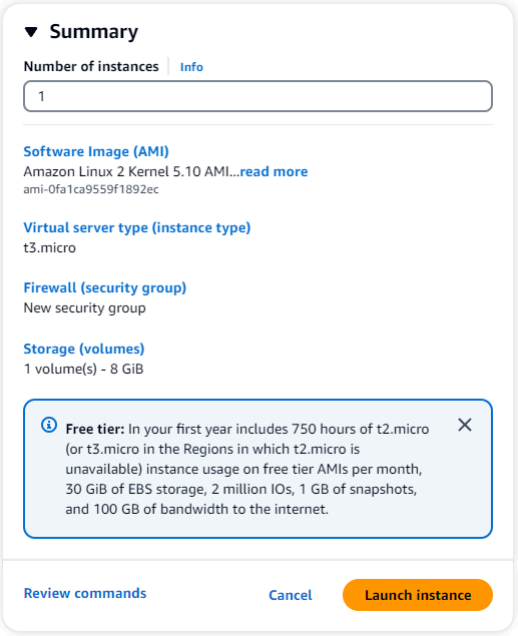
6. Optional: Wählen Sie im Abschnitt Netzwerkeinstellungen unter Firewall (Sicherheitsgruppen) entweder Sicherheitsgruppe erstellen oder Vorhandene Sicherheitsgruppe auswählen aus.



   Wenn Sie eine vorhandene Sicherheitsgruppe auswählen möchten, wählen Sie eine aus der Dropdown-Liste Allgemeine Sicherheitsgruppen aus.

   Wenn Sie eine neue Sicherheitsgruppe erstellen möchten, überprüfen Sie alle Regeln für die Zugriffserlaubnis, die für Ihre EC2-Konnektivität gelten.

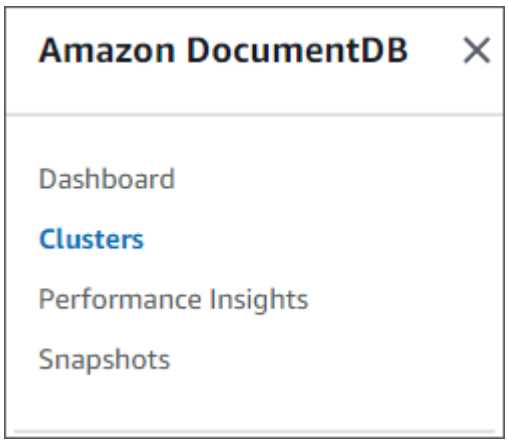
7. Überprüfen Sie im Abschnitt Zusammenfassung Ihre EC2-Konfiguration und wählen Sie Instance starten aus, falls korrekt.Bearbeiten Sie Sicherheitsgruppen.



Schritt 2: Erstellen eines Amazon DocumentDB-Clusters

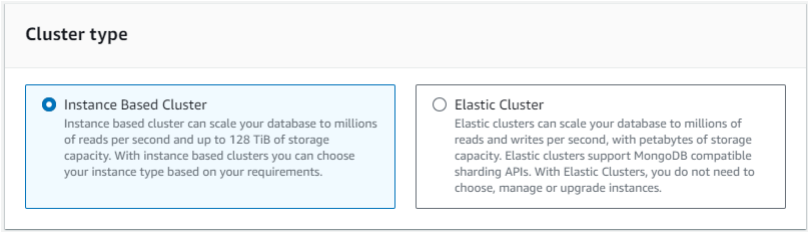
Während die Amazon EC2-Instance bereitgestellt wird, erstellen Sie Ihren Amazon DocumentDB-Cluster.

1. Navigieren Sie zur Amazon DocumentDB-Konsole und wählen Sie im Navigationsbereich Cluster aus.

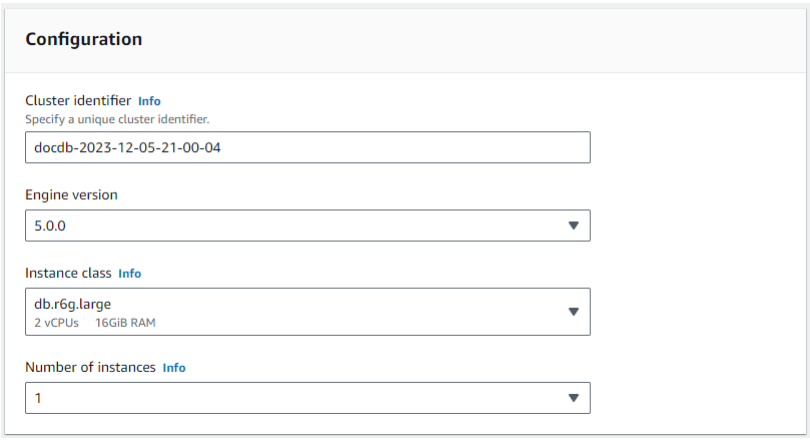


2. Wählen Sie Erstellen.

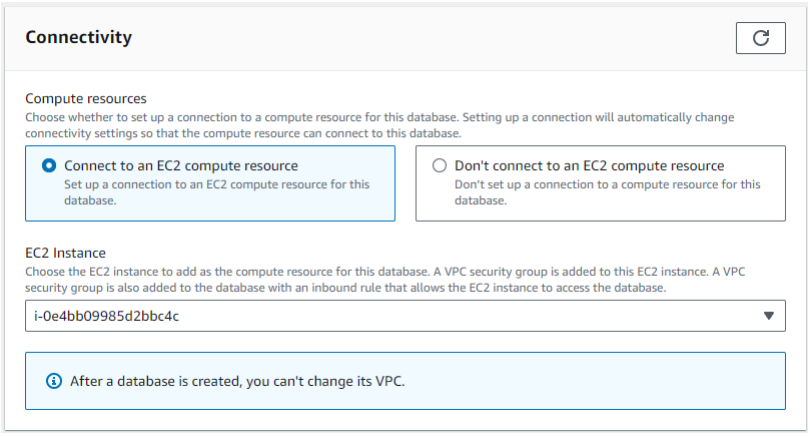
3. Belassen Sie die Einstellung Cluster-Typ auf der Standardeinstellung Instance-basierter Cluster .



4. Wählen Sie für Anzahl der Instances 1 aus. Dadurch werden die Kosten minimiert. Belassen Sie die anderen Einstellungen auf ihrer Standardeinstellung.



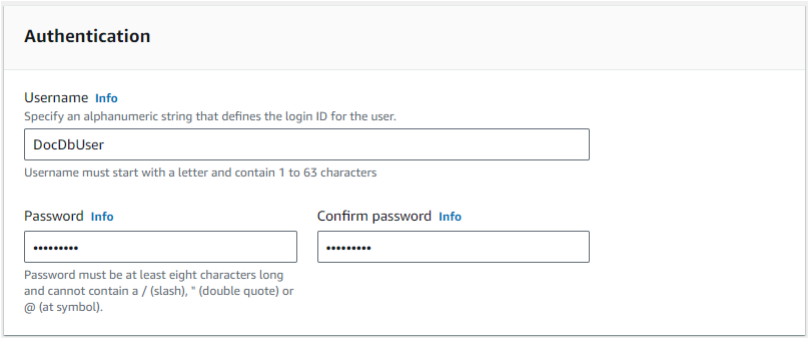
5. Wählen Sie für Konnektivität die Option Mit einer EC2-Rechenressource verbinden aus. Dies ist die EC2-Instance, die Sie in Schritt 1 erstellt haben.



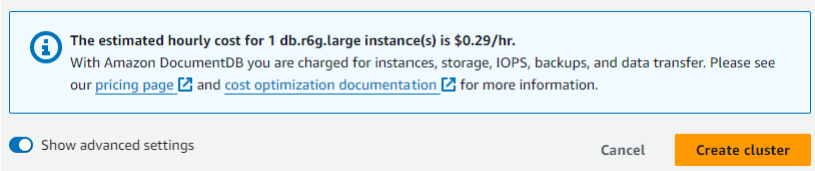
> **Note**
>
> Durch die Verbindung mit einer EC2-Rechenressource wird automatisch eine Sicherheitsgruppe für Ihre EC2-Rechenressourcenverbindung mit Ihrem Amazon DocumentDB-Cluster erstellt. Wenn Sie die Erstellung Ihres Clusters abgeschlossen haben und die neu erstellte Sicherheitsgruppe sehen möchten, navigieren Sie zur Clusterliste und wählen Sie die Kennung Ihres

Clusters aus. Gehen Sie auf der Registerkarte Konnektivität und Sicherheit zu Sicherheitsgruppen und suchen Sie Ihre Gruppe unter Sicherheitsgruppenname (ID). Es enthält den Namen Ihres Clusters und sieht in etwa wie folgt aus: `docdb-ec2-docdb-2023-12-11-21-33-41:i-0e4bb09985d2bbc4c (sg-0238e0b0bf0f73877)`.

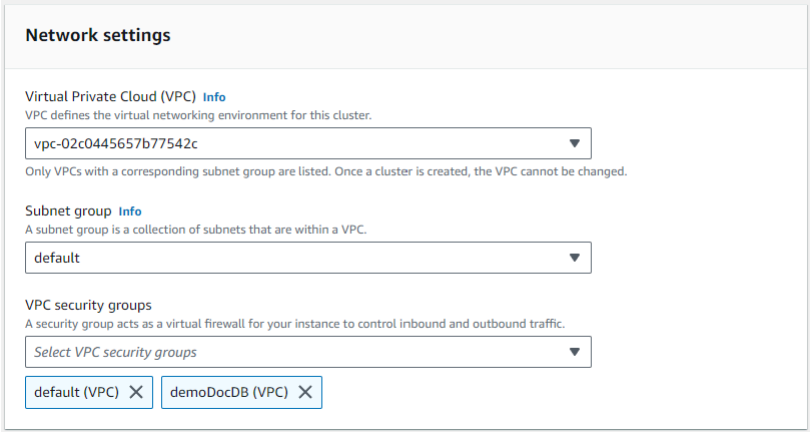
6. Geben Sie für Authentifizierung Anmeldeinformationen ein. Wichtig: Sie benötigen die Anmeldeinformationen, um Ihren Cluster in einem späteren Schritt zu authentifizieren.



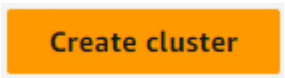
7. Aktivieren Sie Erweiterte Einstellungen anzeigen.



8. Wählen Sie im Abschnitt Netzwerkeinstellungen für Amazon-VPC-Sicherheitsgruppen die Option demoDocDB aus.



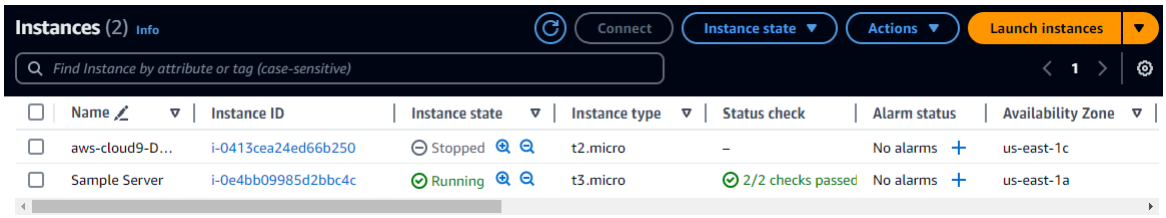
9. Wählen Sie Cluster erstellen.



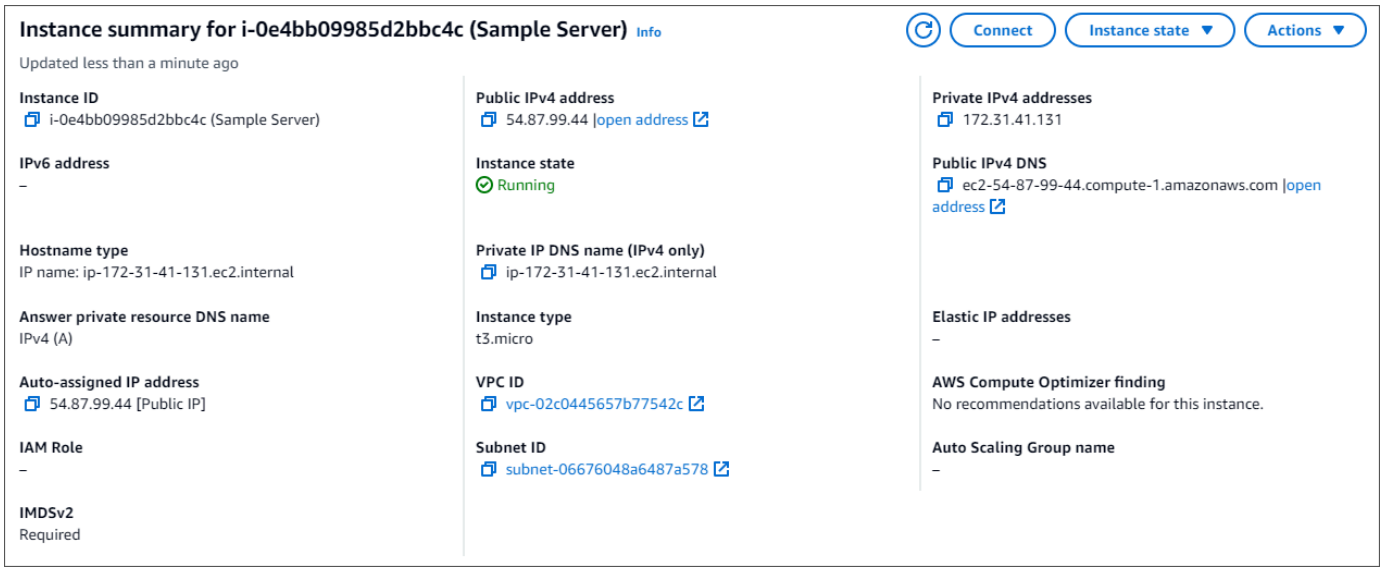
Schritt 3: Herstellen einer Verbindung mit Ihrer Amazon EC2-Instance

Um die mongo-Shell zu installieren, müssen Sie zunächst eine Verbindung zu Ihrer Amazon EC2 herstellen. Durch die Installation der mongo-Shell können Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen und diesen abfragen. Führen Sie folgende Schritte aus:

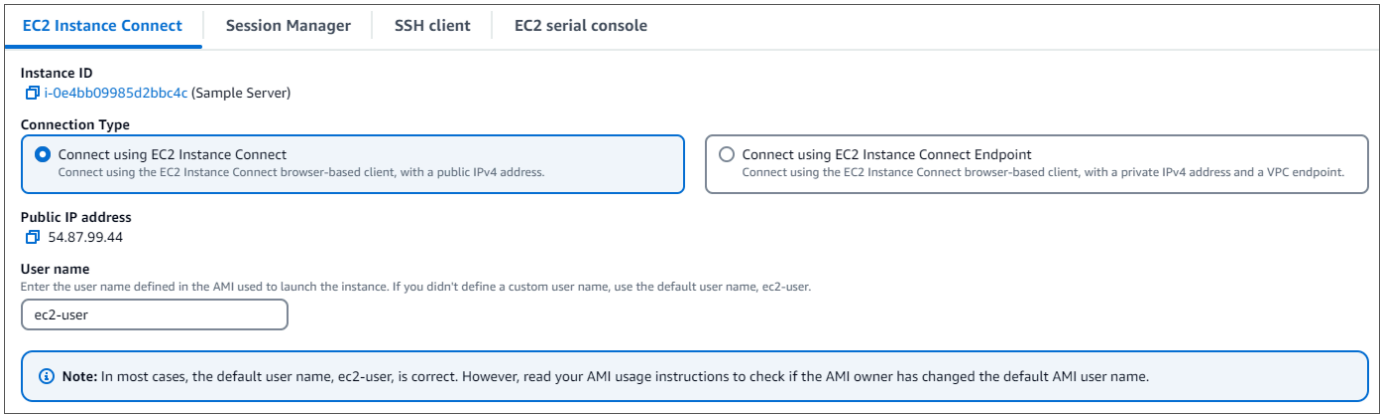
1. Navigieren Sie in der Amazon EC2-Konsole zu Ihren Instances und prüfen Sie, ob die gerade erstellte Instance ausgeführt wird. Wenn dies der Fall ist, wählen Sie die Instance aus, indem Sie auf die Instance-ID klicken.



2. Wählen Sie Connect aus.



3. Es gibt vier Optionen mit Registerkarten für Ihre Verbindungsmethode: Amazon EC2 Instance Connect, Session Manager, SSH-Client oder serielle EC2-Konsole. Sie müssen eine auswählen und deren Anweisungen folgen. Wenn Sie fertig sind, wählen Sie Verbinden aus.



> **Note**
>
> Wenn sich Ihre IP-Adresse geändert hat, nachdem Sie diese Anleitung gestartet haben, oder Sie zu einem späteren Zeitpunkt wieder in Ihre Umgebung zurückkehren, müssen Sie Ihre demoEC2 Sicherheitsgruppenregel für eingehenden Datenverkehr aktualisieren, um eingehenden Datenverkehr von Ihrer neuen API-Adresse zu ermöglichen.

Schritt 4: Installieren der mongo-Shell

Sie können jetzt die mongo-Shell installieren. Dabei handelt es sich um ein Befehlszeilendienstprogramm, mit dem Sie Ihren Amazon DocumentDB-Cluster verbinden und abfragen. Befolgen Sie die folgenden Anweisungen, um die mongo-Shell für Ihr Betriebssystem zu installieren.

On Amazon Linux

So installieren Sie die mongo-Shell auf Amazon Linux

1. Erstellen Sie die Repository-Datei. Führen Sie in der Befehlszeile Ihrer EC2-Instance den folgenden Befehl aus:

   ```
   echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
   repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
    \ngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/
   yum.repos.d/mongodb-org-5.0.repo
   ```

2. Wenn der Vorgang abgeschlossen ist, installieren Sie die mongo-Shell, indem Sie den folgenden Befehl ausführen:

```
sudo yum install -y mongodb-org-shell
```

On Ubuntu 18.04

So installieren Sie die mongo-Shell auf Ubuntu

1. Importieren Sie den öffentlichen Schlüssel, der von dem Paketverwaltungssystem verwendet wird.

   ```
   sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
    2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
   ```

2. Erstellen Sie die Listendatei `/etc/apt/sources.list.d/mongodb-org-3.6.list` für MongoDB mit dem korrekten Befehl für Ihre Ubuntu-Version.

   Ubuntu 18.04

   ```
   echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/
   mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
   org-3.6.list
   ```

   > **Note**
   >
   > Der obige Befehl installiert die mongo 3.6 Shell für Bionic und Xenial.

3. Laden Sie die lokale Paketdatenbank mit dem folgenden Befehl neu:

   ```
   sudo apt-get update
   ```

4. Installieren der MongoDB-Shell.

   ```
   sudo apt-get install -y mongodb-org-shell
   ```

Weitere Informationen zum Installieren von früheren Versionen von MongoDB auf Ihrem Ubuntu-System finden Sie unter Installieren von MongoDB Community Edition auf Ubuntu.

On other operating systems

Informationen zum Installieren der mongo-Shell auf anderen Betriebssystemen finden Sie unter Installieren von MongoDB Community Edition in der MongoDB-Dokumentation.

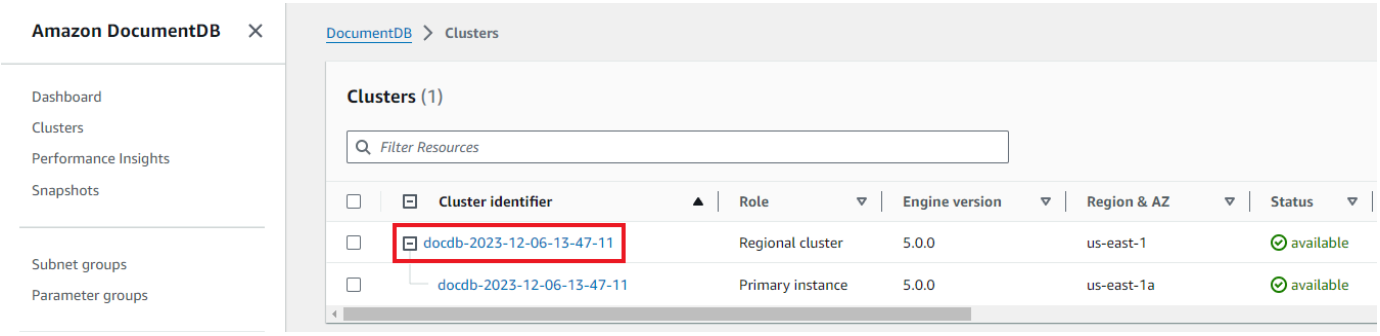
Schritt 5: Verwalten von Amazon DocumentDB TLS

Laden Sie das CA-Zertifikat für Amazon DocumentDB mit dem folgenden Code herunter: `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

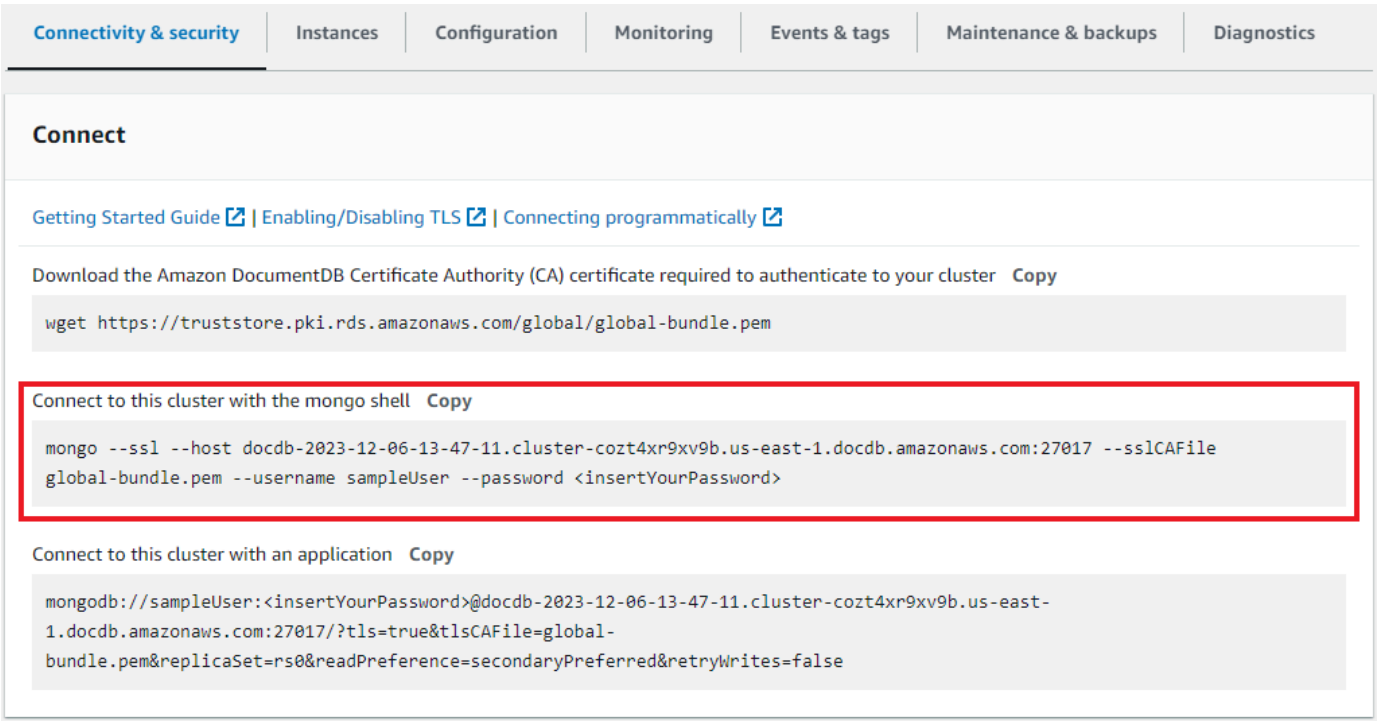
> **Note**
>
> Transport Layer Security (TLS) ist standardmäßig für alle neuen Amazon DocumentDB-Cluster aktiviert. Weitere Informationen finden Sie unter Verwalten der TLS-Einstellungen für Amazon DocumentDB-Cluster.

Schritt 6: Herstellen einer Verbindung mit Ihrem Amazon DocumentDB-Cluster

1. Suchen Sie in der Amazon DocumentDB-Konsole unter Cluster Ihren Cluster. Wählen Sie den Cluster aus, den Sie erstellt haben, indem Sie auf die Cluster-ID klicken.



2. Suchen Sie auf der Registerkarte Konnektivität und Sicherheit im Feld Verbinden die Option Verbindung zu diesem Cluster mit der mongo-Shell herstellen:



Kopieren Sie die angegebene Verbindungszeichenfolge und fügen Sie sie in Ihr Terminal ein.

Nehmen Sie die folgenden Änderungen vor:

a. Stellen Sie sicher, dass Sie den richtigen Benutzernamen in der Zeichenfolge haben.

b. Lassen Sie es weg, `<insertYourPassword>` damit Sie von der mongo-Shell beim Herstellen einer Verbindung zur Eingabe des Passworts aufgefordert werden.

Ihre Verbindungszeichenfolge sollte in etwa wie folgt aussehen:

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem
--username demoUser --password
```

3. Drücken Sie die Eingabetaste in Ihrem Terminal. Sie werden jetzt zur Eingabe Ihres Passworts aufgefordert. Geben Sie Ihr Passwort ein.
4. Wenn Sie Ihr Passwort eingeben und die `rs0:PRIMARY>` Eingabeaufforderung sehen können, sind Sie erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden.

Haben Sie Probleme mit der Verbindung? Weitere Informationen finden Sie unter Fehlerbehebung bei Amazon DocumentDB.

Schritt 7: Einfügen und Abfragen von Daten

Nachdem Sie nun mit Ihrem Cluster verbunden sind, können Sie einige Abfragen ausführen, um sich mit der Verwendung einer Dokumentdatenbank vertraut zu machen.

1. Um ein einzelnes Dokument einzufügen, geben Sie Folgendes ein:

```
db.collection.insert({"hello":"DocumentDB"})
```

2. Sie erhalten die folgende Ausgabe:

```
WriteResult({ "nInserted" : 1 })
```

3. Sie können das Dokument lesen, das Sie mit dem `findOne()` Befehl geschrieben haben (da es nur ein einzelnes Dokument zurückgibt). Geben Sie Folgendes ein:

```
db.collection.findOne()
```

4. Sie erhalten die folgende Ausgabe:

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" :
"DocumentDB" }
```

5. Um einige weitere Abfragen durchzuführen, sollten Sie einen Anwendungsfall für Spieleprofile in Betracht ziehen. Fügen Sie zunächst einige Einträge in eine Sammlung mit dem Titel ein`profiles`. Geben Sie Folgendes ein:

```
db.profiles.insertMany([
           { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
           { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,
 "score":9},
           { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
 "score":87},
           { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
 "score":27}
           ])
```

6. Sie erhalten die folgende Ausgabe:

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. Verwenden Sie den `find()` Befehl , um alle Dokumente in der Profilsammlung zurückzugeben. Geben Sie Folgendes ein:

```
db.profiles.find()
```

8. Sie erhalten eine Ausgabe, die mit den Daten übereinstimmt, die Sie in Schritt 5 eingegeben haben.
9. Verwenden Sie eine Abfrage für ein einzelnes Dokument mithilfe eines Filters. Geben Sie Folgendes ein:

```
db.profiles.find({name: "Katie"})
```

10. Sie sollten diese Ausgabe zurückgeben:

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
```

11. Versuchen wir nun, ein Profil zu finden und es mit dem `findAndModify` Befehl zu ändern. Wir geben dem Benutzer zusätzliche zehn Punkte mit dem folgenden Code:

```
db.profiles.findAndModify({
    query: { name: "Matt", status: "active"},
    update: { $inc: { score: 10 } }
})
```

12. Sie erhalten die folgende Ausgabe (beachten Sie, dass sein Wert noch nicht gestiegen ist):

```
{
"_id" : 1,
"name" : "Matt",
"status" : "active",
"level" : 12,
"score" : 202
}
```

13. Mit der folgenden Abfrage können Sie überprüfen, ob sich sein Wert geändert hat:

```
db.profiles.find({name: "Matt"})
```

14. Sie erhalten die folgende Ausgabe:

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12,
"score" : 212 }
```

Schritt 8: Erkunden

Herzlichen Glückwunsch! Sie haben die Schnellstartanleitung für Amazon DocumentDB erfolgreich abgeschlossen.

Was ist als Nächstes? Erfahren Sie, wie Sie diese leistungsstarke Datenbank mit einigen ihrer beliebten Funktionen vollständig nutzen können:

- Verwalten von Amazon DocumentDB
- Skalierung
- Sichern und Wiederherstellen

> **Note**
>
> Um Kosten zu sparen, können Sie entweder Ihren Amazon DocumentDB-Cluster stoppen, um die Kosten zu senken, oder den Cluster löschen. Standardmäßig stoppt Ihre AWS Cloud9 Umgebung nach 30 Minuten Inaktivität die zugrunde liegende Amazon EC2.

Automatisches Verbinden einer EC2-Instance mit einer vorhandenen Amazon DocumentDB-Datenbank

Im folgenden Verfahren wird davon ausgegangen, dass Sie über einen vorhandenen Amazon DocumentDB-Cluster und eine vorhandene Amazon EC2 verfügen.

Zugriff auf Ihren Amazon DocumentDB-Cluster und Einrichten der Amazon EC2Verbindung

1. Greifen Sie auf Ihren Amazon DocumentDB-Cluster zu.

    a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

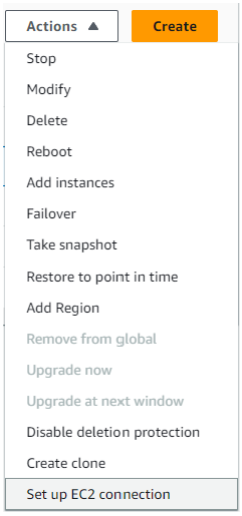
    b. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

c. Geben Sie den gewünschten Cluster an, indem Sie die Schaltfläche links neben dem Namen des Clusters auswählen.

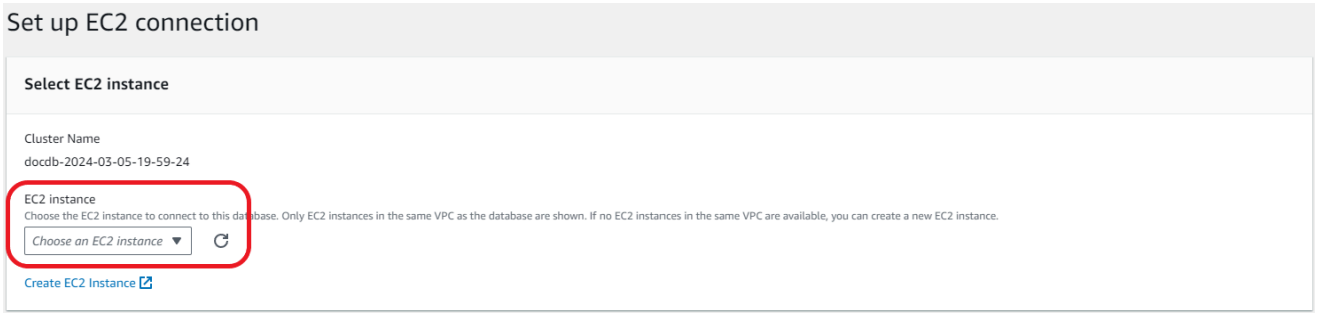
2. Richten Sie die Amazon EC2-Verbindung ein.

a. Wählen Sie Aktionen und dann ECEC2Verbindung einrichten aus.



Das Dialogfeld EC2-Verbindung einrichten wird angezeigt.

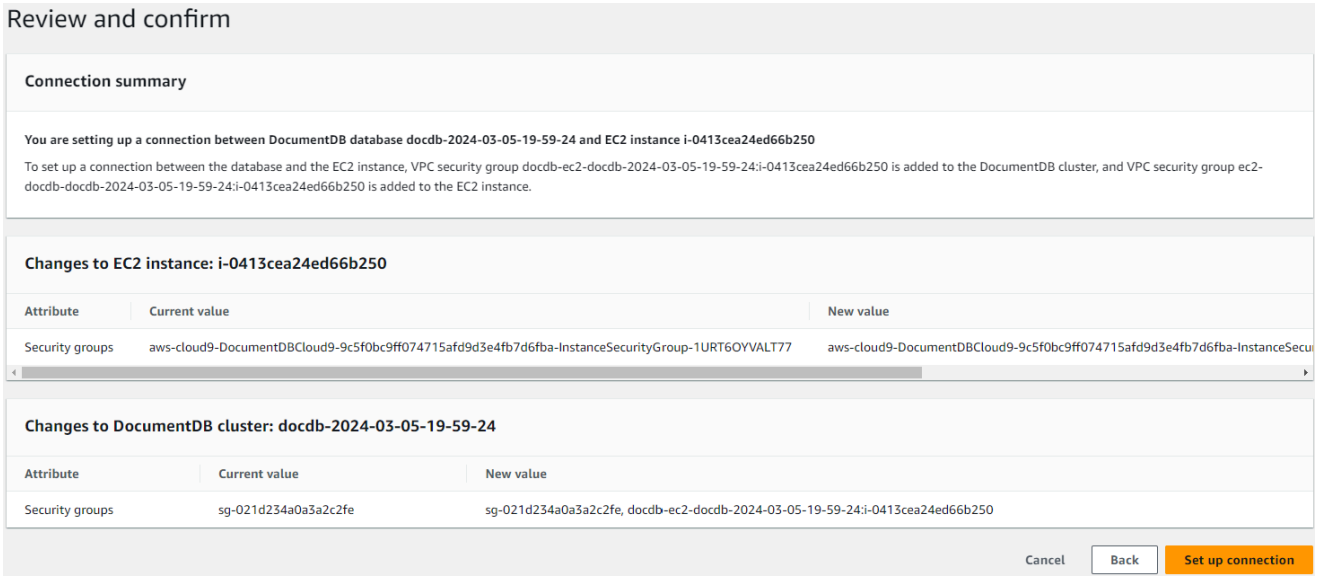
b. Wählen Sie im Feld EC2-Instance die EC2-Instance aus, die Sie mit Ihrem Cluster verbinden möchten.



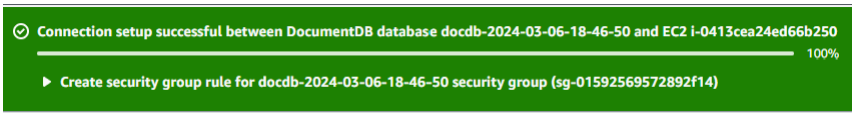
c. Klicken Sie auf Weiter.

Das Dialogfeld Überprüfen und bestätigen wird angezeigt.

d. Stellen Sie sicher, dass die Änderungen korrekt sind. Wählen Sie dann Verbindung einrichten aus.



Bei Erfolg wird die folgende Überprüfung angezeigt:



Übersicht über die automatische Verbindung mit einer EC2-Instance

Wenn Sie eine Verbindung zwischen einer EC2-Instance und einer Amazon DocumentDB-Datenbank einrichten, konfiguriert Amazon DocumentDB automatisch die VPC-Sicherheitsgruppe für Ihre EC2-Instance und für Ihre Amazon DocumentDB-Datenbank.

Im Folgenden sind die Anforderungen für die Verbindung einer EC2-Instance mit einer Amazon DocumentDB-Datenbank aufgeführt:

- Die EC2-Instance muss sich in derselben VPC wie die Amazon DocumentDB-Datenbank befinden.

  Wenn keine EC2-Instances in derselben VPC vorhanden sind, dann bietet die Konsole einen Link zum Erstellen einer solchen Instance.
- Der Benutzer, der die Verbindung einrichtet, muss über Berechtigungen zum Ausführen der folgenden Amazon-EC2-Vorgänge verfügen:
  - `ec2:AuthorizeSecurityGroupEgress`

- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Auf Ihrem Konto fallen ggf. Kosten über Availability Zones hinweg an, wenn sich die DB-Instance und die EC2-Instance in unterschiedlichen Availability Zones befinden.

Wenn Sie eine Verbindung zu einer EC2-Instance einrichten, handelt Amazon DocumentDB gemäß der aktuellen Konfiguration der Sicherheitsgruppen, die der Amazon DocumentDB-Datenbank und der EC2-Instance zugeordnet sind, wie in der folgenden Tabelle beschrieben:

Aktuelle Konfiguration der Amazon DocumentDB-Sicherheitsgruppe	Aktuelle EC2-Sicherheitsgruppenkonfiguration	Amazon DocumentDB-Aktion
Der Amazon DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich t`DocumentDB-ec2-n` . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle.	Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht `DocumentDB-ec2-n` (wobei n eine Zahl ist). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Amazon DocumentDB-Datenbank als Quelle.	Amazon DocumentDB ergreift keine Maßnahmen. Es wurde bereits automatisch eine Verbindung zwischen der EC2-Instance und der Amazon DocumentDB-Datenbank konfiguriert. Da bereits eine Verbindung zwischen der EC2-Instance und der Amazon DocumentDB-Datenbank besteht, werden die Sicherheitsgruppen nicht geändert.

Aktuelle Konfiguration der Amazon DocumentDB-Sicherheitsgruppe	Aktuelle EC2-Sicherheitsgruppenkonfiguration	Amazon DocumentDB-Aktion
Es gilt eine der folgenden Bedingungen: • Der Amazon DocumentDB-Datenbank ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entsprich tDocumentDB-ec2-n . • Der Amazon DocumentDB sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich tDocumentDB-ec2-n. Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2-Instance verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle enthält. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde. Beispiele für Änderungen sind das Hinzufügen einer Regel	Es gilt eine der folgenden Bedingungen: • Der EC2-Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster ec2-DocumentDB-n entspricht. • Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster ec2-DocumentDB-n entspricht. Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Amazon DocumentDB-Datenbank verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Amazon DocumentDB-Datenbank als Quelle enthält. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Amazon DocumentDB-Aktion: Erstellen neuer Sicherheitsgruppen

Aktuelle Konfiguration der Amazon DocumentDB-Sicherheitsgruppe	Aktuelle EC2-Sicherheitsgruppenkonfiguration	Amazon DocumentDB-Aktion
oder das Ändern des Ports einer vorhandenen Regel.		
Der Amazon DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich t`DocumentDB-ec2-n` . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle.	Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `ec2-DocumentDB-n` entspricht. Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Amazon DocumentDB-Datenbank verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Amazon DocumentDB-Datenbank als Quelle enthält. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Amazon DocumentDB-Aktion: Erstellen neuer Sicherheitsgruppen

Aktuelle Konfiguration der Amazon DocumentDB-Sicherheitsgruppe	Aktuelle EC2-Sicherheitsgruppenkonfiguration	Amazon DocumentDB-Aktion
Der Amazon DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich `tDocumentDB-ec2-n` . Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle.	Eine gültige EC2-Sicherheitsgruppe für die Verbindung ist vorhanden, jedoch nicht mit der EC2-Instance verknüpft. Die Sicherheitsgruppe trägt einen Namen, der dem Muster `DocumentDB-ec2-n` entspricht. Sie wurde nicht geändert. Sie enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Amazon DocumentDB-Datenbank als Quelle.	Amazon DocumentDB-Aktion: EC2-Sicherheitsgruppe zuordnen

Aktuelle Konfiguration der Amazon DocumentDB-Sicherheitsgruppe	Aktuelle EC2-Sicherheitsgruppenkonfiguration	Amazon DocumentDB-Aktion
Es gilt eine der folgenden Bedingungen: • Der Amazon DocumentDB-Datenbank ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster entsprich t`DocumentDB-ec2-n` . • Der Amazon DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprich t`DocumentDB-ec2-n`. Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2-Instance verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle enthält. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Der EC2-Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe enthält nur eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Amazon DocumentDB-Datenbank als Quelle.	Amazon DocumentDB-Aktion: Erstellen neuer Sicherheitsgruppen

Amazon DocumentDB-Aktion: Erstellen neuer Sicherheitsgruppen

Amazon DocumentDB führt die folgenden Aktionen aus:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `DocumentDB-ec2-n` entspricht. Diese Sicherheitsgruppe enthält eine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2-Instance als Quelle. Diese Sicherheitsgruppe ist der Amazon DocumentDB-Datenbank zugeordnet und ermöglicht der EC2-Instance den Zugriff auf die Amazon DocumentDB-Datenbank.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `ec2-DocumentDB-n` entspricht. Diese Sicherheitsgruppe enthält eine Regel für ausgehenden Datenverkehr mit der VPC-Sicherheitsgruppe der Amazon DocumentDB-Datenbank als Quelle. Diese Sicherheitsgruppe ist der EC2-Instance zugeordnet und ermöglicht es der EC2-Instance, Datenverkehr an die Amazon DocumentDB-Datenbank zu senden.

Amazon DocumentDB-Aktion: EC2-Sicherheitsgruppe zuordnen

Amazon DocumentDB ordnet die gültige, vorhandene EC2-Sicherheitsgruppe der EC2-Instance zu. Diese Sicherheitsgruppe ermöglicht es der EC2-Instance, Datenverkehr an die Amazon DocumentDB-Datenbank zu senden.

Anzeigen verbundener Rechenressourcen

Sie können die verwenden AWS Management Console , um die Rechenressourcen anzuzeigen, die mit einer Amazon DocumentDB-Datenbank verbunden sind. Zu den angezeigten Ressourcen gehören Rechenressourcenverbindungen, die automatisch eingerichtet wurden. Sie können die Konnektivität mit Rechenressourcen auf folgende Weise automatisch einrichten:

- Sie können die Rechenressource auswählen, wenn Sie die Datenbank erstellen. Weitere Informationen finden Sie unter Erstellen eines Amazon DocumentDB-Clusters und Erstellen eines Multi-AZ-DB-Clusters.
- Sie können die Konnektivität zwischen einer vorhandenen Datenbank und einer Rechenressource einrichten. Weitere Informationen finden Sie unter Automatisches Verbinden von Amazon EC2 .

Die aufgelisteten Rechenressourcen enthalten keine Ressourcen, die manuell mit der Datenbank verbunden wurden. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf eine Datenbank erlauben, indem Sie der VPC-Sicherheitsgruppe, die der Datenbank zugeordnet ist, eine Regel hinzufügen.

Für die Auflistung einer Rechenressource müssen die folgenden Bedingungen erfüllt sei:

- Der Name der Sicherheitsgruppe, die der Rechenressource zugeordnet ist, entspricht dem Muster `ec2-DocumentDB-n` (wobei n eine Zahl ist).
- Die Sicherheitsgruppe, die der Rechenressource zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei der Portbereich auf den Port festgelegt ist, den die Amazon DocumentDB-Datenbank verwendet.
- Die Sicherheitsgruppe, die der Rechenressource zugeordnet ist, hat eine Regel für ausgehenden Datenverkehr, wobei die Quelle auf eine Sicherheitsgruppe festgelegt ist, die der Amazon DocumentDB-Datenbank zugeordnet ist.
- Der Name der Sicherheitsgruppe, die der Amazon DocumentDB-Datenbank zugeordnet ist, entspricht dem Muster `DocumentDB-ec2-n` (wobei n eine Zahl ist).
- Die Sicherheitsgruppe, die der Amazon DocumentDB-Datenbank zugeordnet ist, hat eine Regel für eingehenden Datenverkehr, wobei der Portbereich auf den Port festgelegt ist, den die Amazon DocumentDB-Datenbank verwendet.
- Die Sicherheitsgruppe, die der Amazon DocumentDB-Datenbank zugeordnet ist, hat eine Regel für eingehenden Datenverkehr, wobei die Quelle auf eine Sicherheitsgruppe festgelegt ist, die der Rechenressource zugeordnet ist.

So zeigen Sie Rechenressourcen an, die mit einer Amazon DocumentDB-Datenbank verbunden sind

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den Namen der Amazon DocumentDB-Datenbank aus.
3. Zeigen Sie auf der Registerkarte Konnektivität und Sicherheit die Rechenressourcen im Abschnitt Verbundene Rechenressourcen an.

Manuelles Verbinden von Amazon EC2

Themen

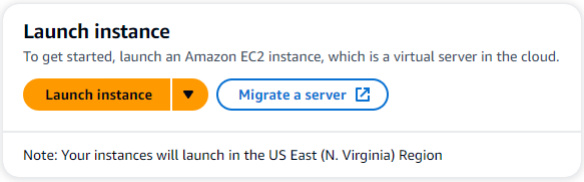
- Schritt 4: Herstellen einer Verbindung mit Ihrer Amazon EC2-Instance
- Schritt 5: Installieren der mongo-Shell
- Schritt 6: Verwalten von Amazon DocumentDB TLS
- Schritt 7: Herstellen einer Verbindung mit Ihrem Amazon DocumentDB-Cluster
- Schritt 8: Einfügen und Abfragen von Daten
- Schritt 9: Erkunden

Bei den folgenden Schritten wird davon ausgegangen, dass Sie die Schritte im Voraussetzungen Thema abgeschlossen haben.

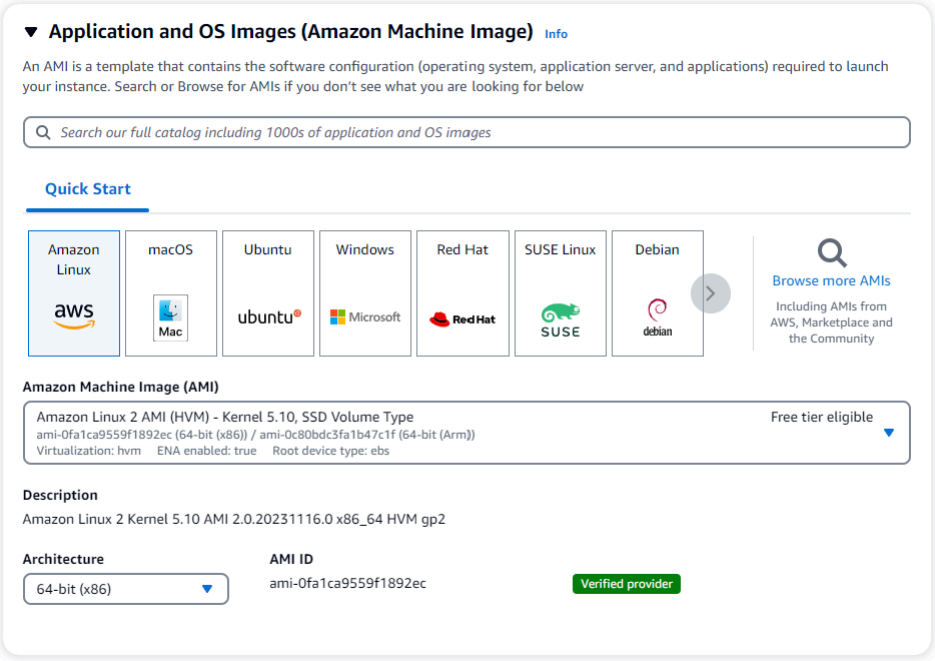
Schritt 1: Erstellen einer Amazon EC2-Instance

In diesem Schritt erstellen Sie eine Amazon EC2-Instance in derselben Region und Amazon VPC, die Sie später zur Bereitstellung Ihres Amazon DocumentDB-Clusters verwenden werden.

1. Wählen Sie in der Amazon EC2-Konsole Instance starten aus.

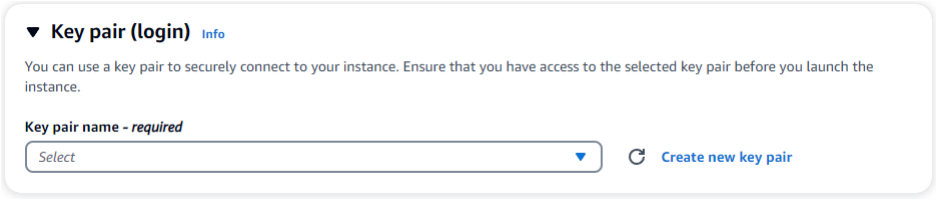


2. Geben Sie einen Namen oder eine Kennung in das Feld Name ein, das sich im Abschnitt Name und Tags befindet.
3. Suchen Sie in der Dropdown-Liste Amazon Machine Image (AMI) nach Amazon Linux 2 AMI und wählen Sie es aus.



4. Suchen Sie t3.micro in der Dropdown-Liste Instance-Typ und wählen Sie es aus.

5. Geben Sie im Abschnitt Schlüsselpaar (Anmeldung) die Kennung eines vorhandenen Schlüsselpaars ein oder wählen Sie Neues Schlüsselpaar erstellen aus.



Sie müssen ein Amazon EC2-Schlüsselpaar angeben.

Wenn Sie über ein Amazon EC2-Schlüsselpaar verfügen:

a. Wählen Sie ein Schlüsselpaar und dann Ihr Schlüsselpaar aus der Liste aus.

b. Sie müssen bereits über die private Schlüsseldatei (PEM- oder PPK-Datei) verfügen, um sich bei Ihrer Amazon EC2-Instance anzumelden.

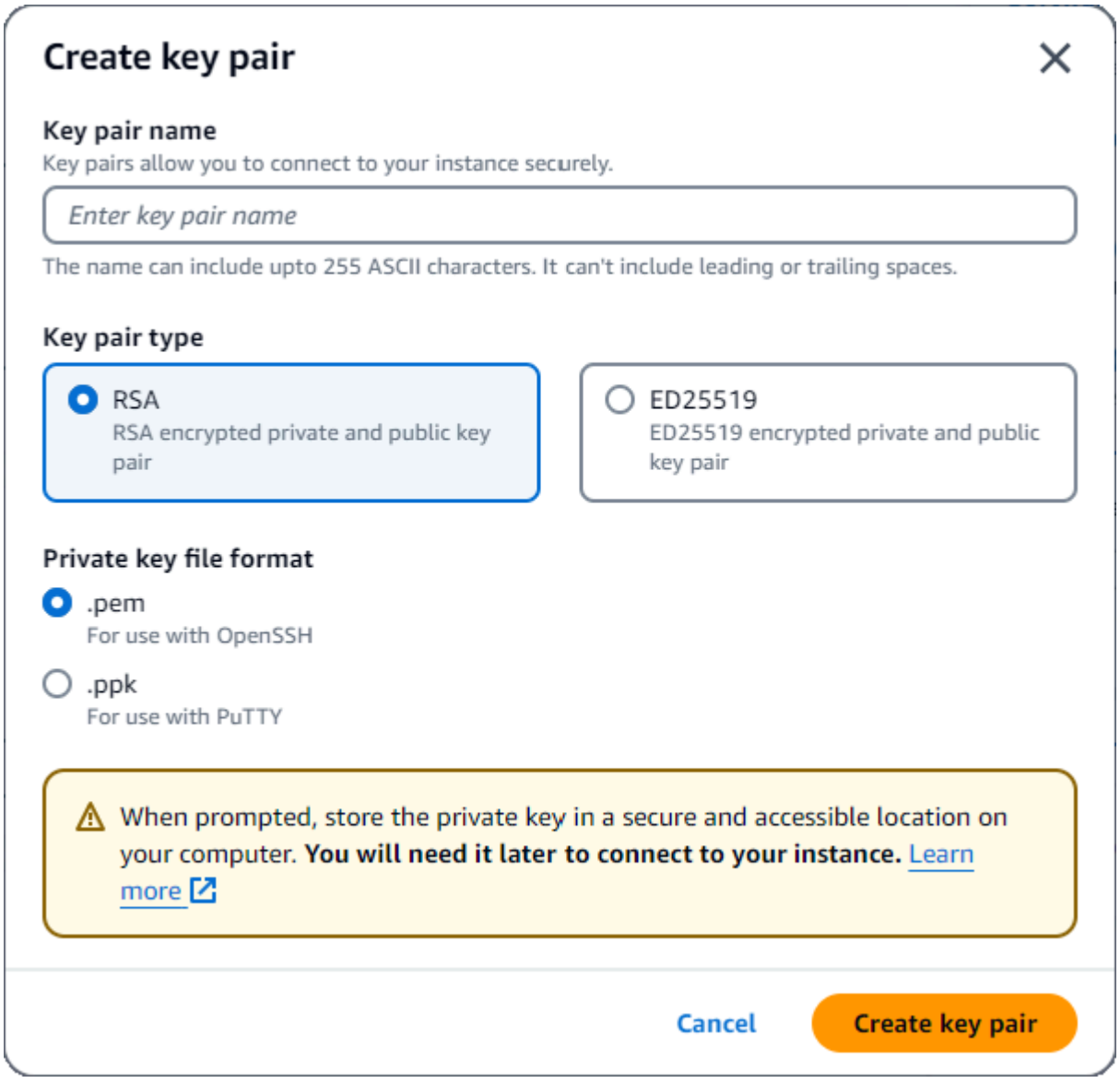
Wenn Sie kein Amazon EC2-Schlüsselpaar haben:

a. Wählen Sie Neues Schlüsselpaar erstellen aus. Daraufhin wird das Dialogfeld Schlüsselpaar erstellen angezeigt.

b. Geben Sie einen Namen in das Feld Schlüsselpaarname ein.

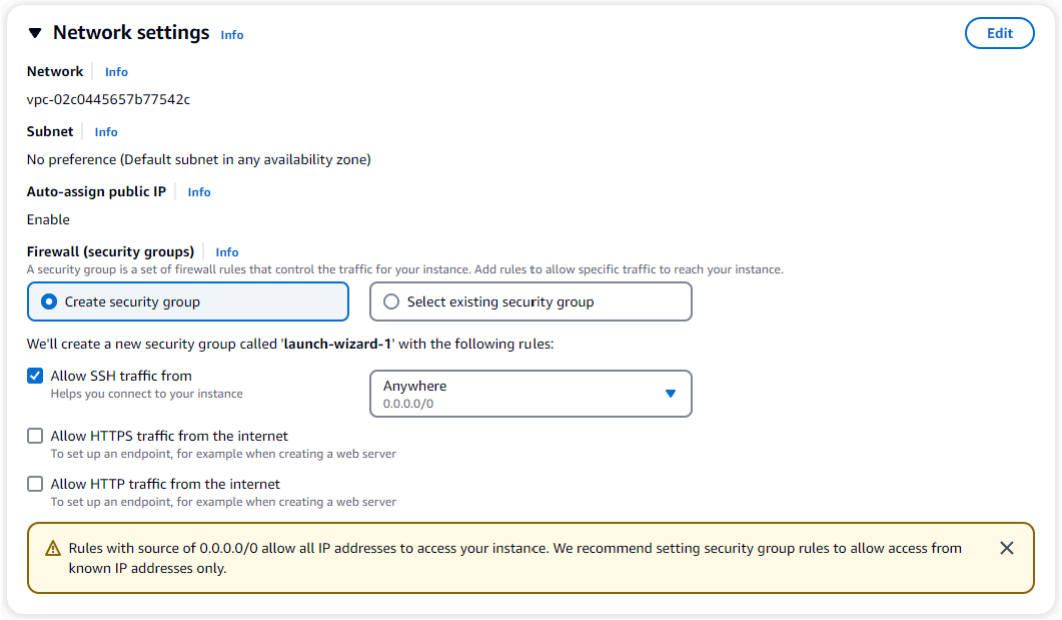
c. Wählen Sie den Schlüsselpaartyp und das Dateiformat für den privaten Schlüssel aus.

d. Wählen Sie Create Key Pair (Schlüsselpaar erstellen) aus.



> **Note**
>
> Aus Sicherheitsgründen empfehlen wir dringend, ein Schlüsselpaar für SSH- und Internetkonnektivität zu Ihrer EC2-Instance zu verwenden.

6. Wählen Sie im Abschnitt Netzwerkeinstellungen unter Firewall (Sicherheitsgruppen) entweder Sicherheitsgruppe erstellen oder Vorhandene Sicherheitsgruppe auswählen aus.



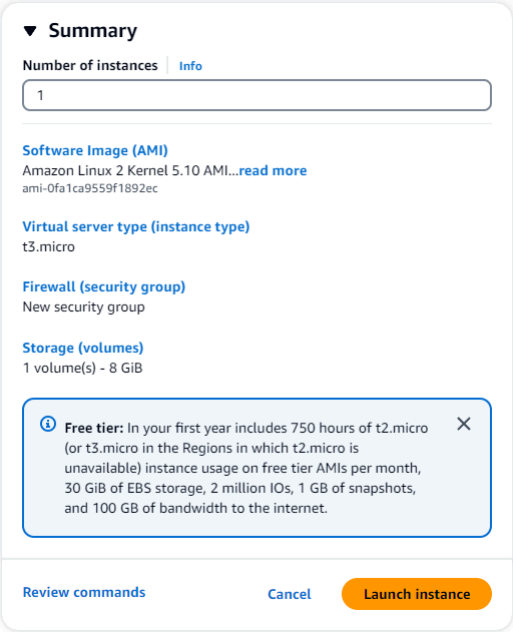
Wenn Sie eine vorhandene Sicherheitsgruppe auswählen möchten, wählen Sie eine aus der Dropdown-Liste Allgemeine Sicherheitsgruppen aus.

Wenn Sie eine neue Sicherheitsgruppe erstellen möchten, gehen Sie wie folgt vor:

a. Überprüfen Sie alle Regeln für das Zulassen von Datenverkehr, die für Ihre EC2-Konnektivität gelten.

b. Wählen Sie im Feld IP die Option Meine IP oder Benutzerdefiniert aus, um aus einer Liste von CIDR-Blöcken, Präfixlisten oder Sicherheitsgruppen auszuwählen. Wir empfehlen Anywhere nicht als Wahl, es sei denn, Ihre EC2-Instance befindet sich in einem isolierten Netzwerk, da dies den IP-Adresszugriff auf Ihre EC2-Instance erlaubt.

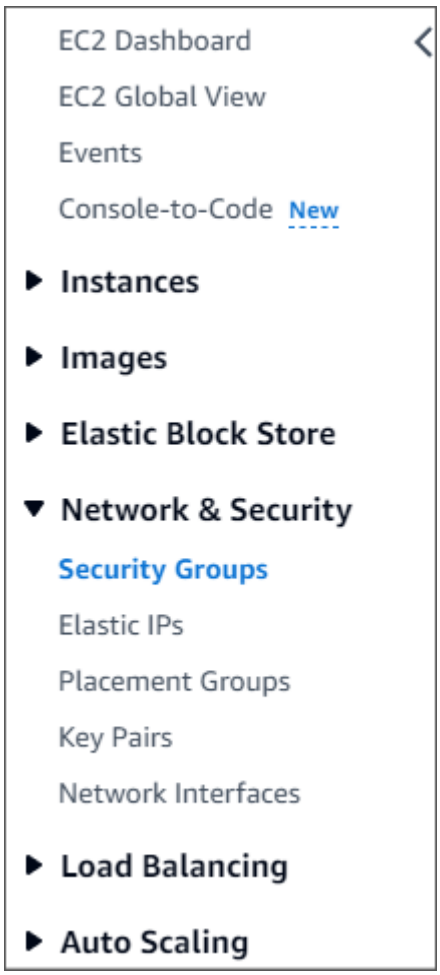
7. Überprüfen Sie im Abschnitt Zusammenfassung Ihre EC2-Konfiguration und wählen Sie Instance starten aus, falls korrekt.Sicherheitsgruppen bearbeiten.



Schritt 2: Erstellen einer Sicherheitsgruppe

Sie erstellen jetzt eine neue Sicherheitsgruppe in Ihrer standardmäßigen Amazon VPC. Mit der Sicherheitsgruppe `demoDocDB` können Sie von Ihrer Amazon DocumentDBAmazon EC2Cluster über Port 27017 (Standardport für Amazon DocumentDB) herstellen.

1. Wählen Sie in der Amazon EC2-Managementkonsole unter Netzwerk und Sicherheit die Option Sicherheitsgruppen aus.



2. Wählen Sie Sicherheitsgruppe erstellen aus.

3. Im Abschnitt Grundlegende Details:

   a. Geben Sie für Security group name (Name der Sicherheitsgruppe) `demoDocDB` ein.

   b. Geben Sie im Feld Description (Beschreibung) eine Beschreibung ein.

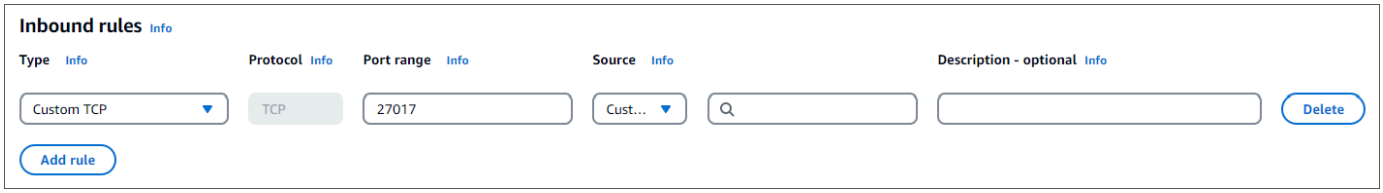
   c. Akzeptieren Sie für VPC die Verwendung Ihrer Standard-VPC.

4. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.

   a. Wählen Sie für Type Custom TCP Rule aus.

   b. Geben Sie für Portbereich ein`27017`.

   c. Wählen Sie für Ziel die Option Benutzerdefiniert aus. Suchen Sie im Feld daneben nach der Sicherheitsgruppe, die Sie gerade mit dem Namen erstellt haben`demoEC2`. Möglicherweise müssen Sie Ihren Browser aktualisieren, damit die Amazon EC2-Konsole den `demoEC2` Quellnamen automatisch ausfüllen kann.

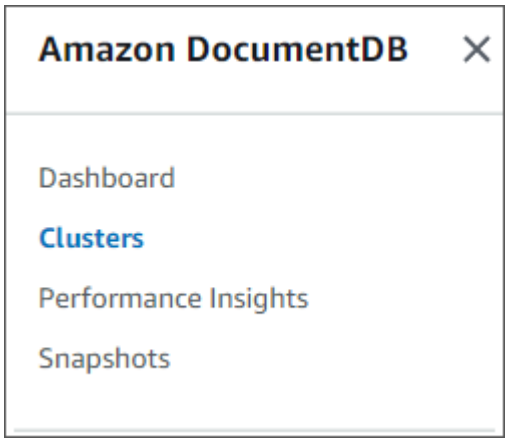


5. Akzeptieren Sie alle anderen Standardwerte und wählen Sie Sicherheitsgruppe erstellen aus.

Schritt 3: Erstellen eines Amazon DocumentDB-Clusters

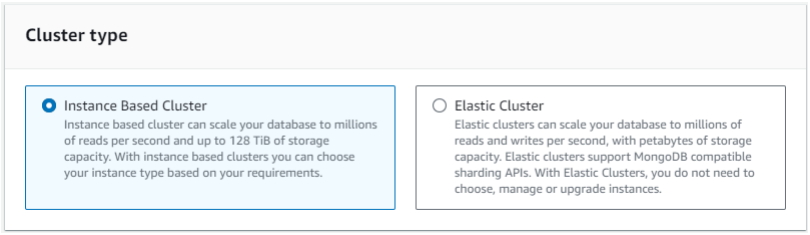
Während die Amazon EC2-Instance bereitgestellt wird, erstellen Sie Ihren Amazon DocumentDB-Cluster.

1. Navigieren Sie zur Amazon DocumentDB-Konsole und wählen Sie im Navigationsbereich Cluster aus.

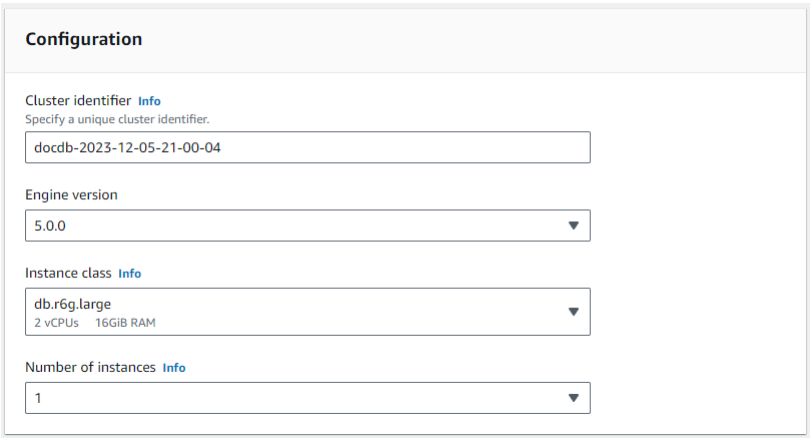


2. Wählen Sie Erstellen.

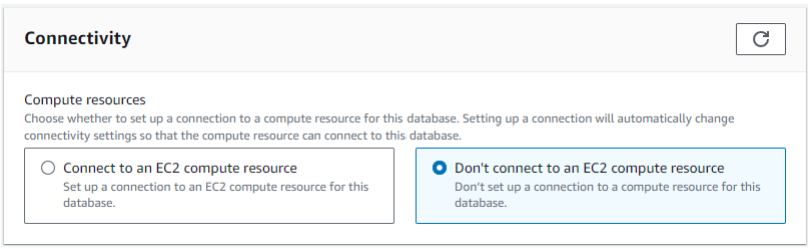
3. Belassen Sie die Einstellung Cluster-Typ auf der Standardeinstellung Instance-basierter Cluster .



4. Wählen Sie für Anzahl der Instances 1 aus. Dadurch werden die Kosten minimiert. Belassen Sie die anderen Einstellungen auf ihrer Standardeinstellung.

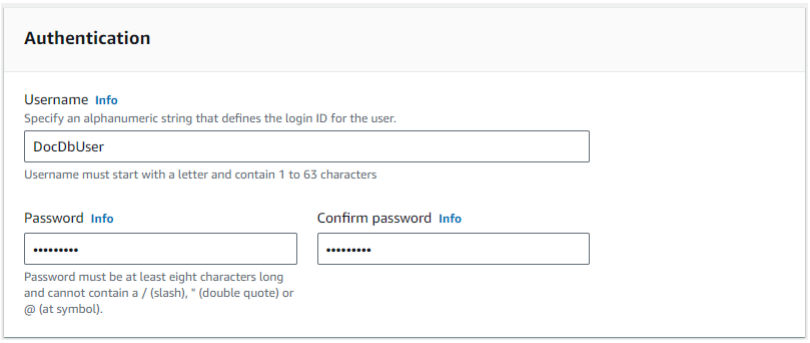


5. Behalten Sie für Konnektivität die Standardeinstellung Keine Verbindung zu einer EC2-Rechenressource herstellt bei.

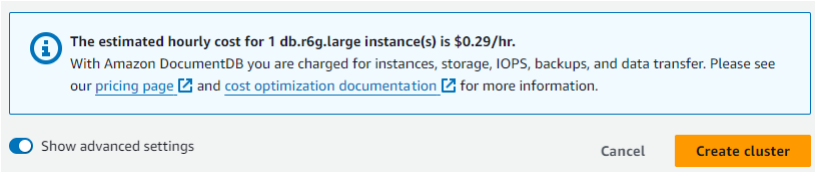


> **Note**
>
> Durch die Verbindung mit einer EC2-Rechenressource werden automatisch Sicherheitsgruppen für Ihre EC2-Rechenressourcenverbindung zu Ihrem Cluster erstellt. Da Sie diese Sicherheitsgruppen im vorherigen Schritt manuell erstellt haben, sollten Sie Keine Verbindung zu einer EC2-Rechenressource herstellen auswählen, um keinen zweiten Satz von Sicherheitsgruppen zu erstellen.

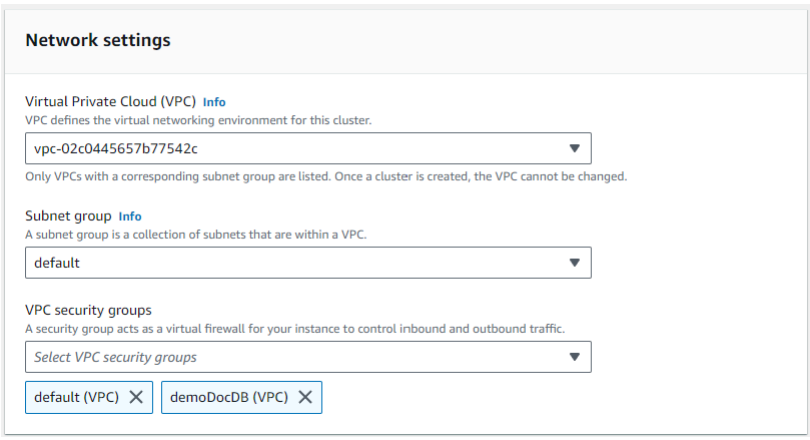
6. Geben Sie für Authentifizierung Anmeldeinformationen ein. Wichtig: Sie benötigen die Anmeldeinformationen, um Ihren Cluster in einem späteren Schritt zu authentifizieren.



7. Aktivieren Sie Erweiterte Einstellungen anzeigen.



8. Wählen Sie im Abschnitt Netzwerkeinstellungen für Amazon-VPC-Sicherheitsgruppen die Option demoDocDB aus.

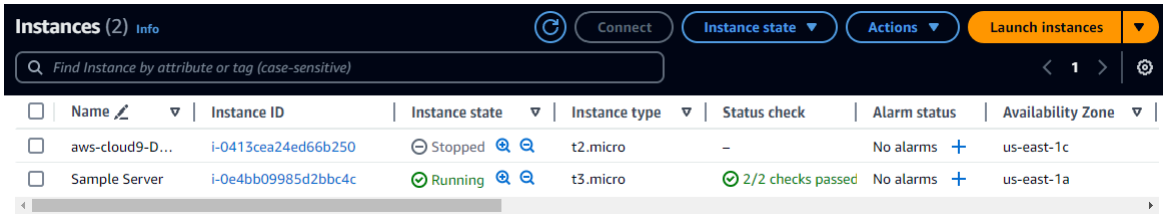


9. Wählen Sie Cluster erstellen.

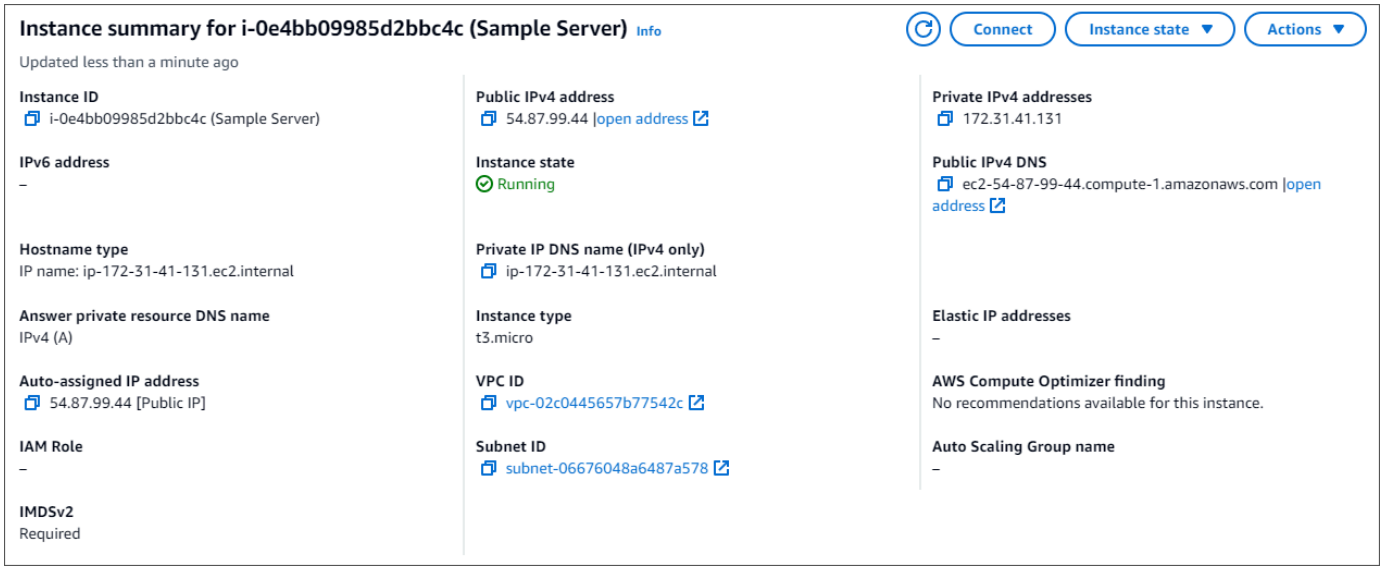
Schritt 4: Herstellen einer Verbindung mit Ihrer Amazon EC2-Instance

Um die mongo-Shell zu installieren, müssen Sie zunächst eine Verbindung zu Ihrer Amazon EC2 herstellen. Durch die Installation der mongo-Shell können Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen und diesen abfragen. Führen Sie folgende Schritte aus:

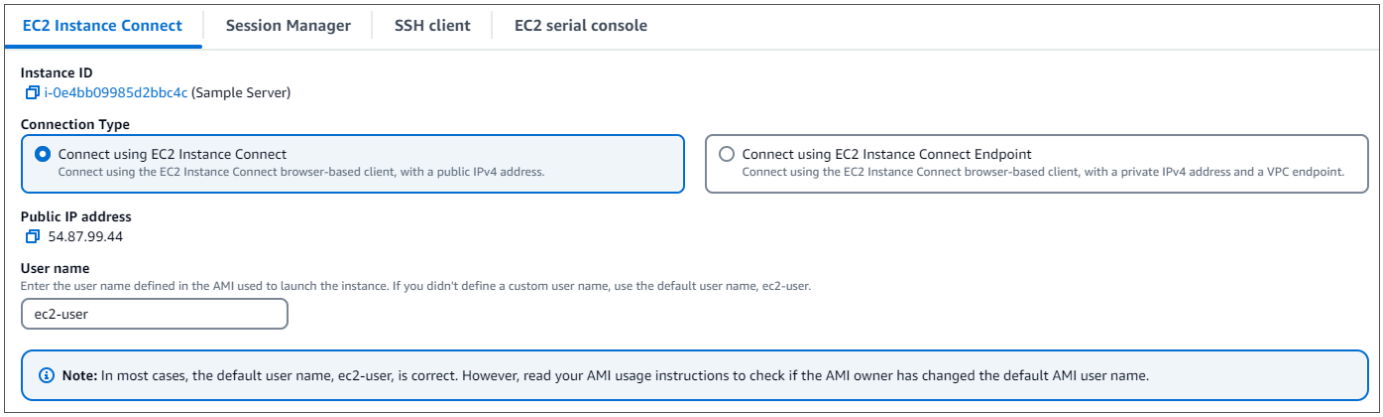
1. Navigieren Sie in der Amazon EC2-Konsole zu Ihren Instances und prüfen Sie, ob die gerade erstellte Instance ausgeführt wird. Wenn dies der Fall ist, wählen Sie die Instance aus, indem Sie auf die Instance-ID klicken.



2. Wählen Sie Connect aus.



3. Es gibt vier Optionen mit Registerkarten für Ihre Verbindungsmethode: Amazon EC2 Instance Connect, Session Manager, SSH-Client oder serielle EC2-Konsole. Sie müssen eine auswählen und deren Anweisungen folgen. Wenn Sie fertig sind, wählen Sie Verbinden aus.



> **Note**
>
> Wenn sich Ihre IP-Adresse geändert hat, nachdem Sie diese Anleitung gestartet haben, oder Sie zu einem späteren Zeitpunkt in Ihre Umgebung zurückkehren, müssen Sie Ihre demoEC2 Sicherheitsgruppenregel für eingehenden Datenverkehr aktualisieren, um eingehenden Datenverkehr von Ihrer neuen API-Adresse zu ermöglichen.

Schritt 5: Installieren der mongo-Shell

Sie können jetzt die mongo-Shell installieren. Dabei handelt es sich um ein Befehlszeilendienstprogramm, mit dem Sie Ihren Amazon DocumentDB-Cluster verbinden und abfragen. Folgen Sie den folgenden Anweisungen, um die mongo-Shell für Ihr Betriebssystem zu installieren.

On Amazon Linux

So installieren Sie die mongo-Shell auf Amazon Linux

1. Erstellen Sie die Repository-Datei. Führen Sie in der Befehlszeile Ihrer EC2-Instance den folgenden Befehl aus:

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc" | sudo tee /etc/
yum.repos.d/mongodb-org-5.0.repo
```

2. Wenn der Vorgang abgeschlossen ist, installieren Sie die mongo-Shell, indem Sie den folgenden Befehl ausführen:

```
sudo yum install -y mongodb-org-shell
```

On Ubuntu 18.04

So installieren Sie die mongo-Shell auf Ubuntu

1. Importieren Sie den öffentlichen Schlüssel, der von dem Paketverwaltungssystem verwendet wird.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. Erstellen Sie die Listendatei `/etc/apt/sources.list.d/mongodb-org-3.6.list` für MongoDB mit dem korrekten Befehl für Ihre Ubuntu-Version.

   Ubuntu 18.04

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/
mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
org-3.6.list
```

> **Note**
>
> Der obige Befehl installiert die mongo 3.6 Shell für Bionic und Xenial.

3. Laden Sie die lokale Paketdatenbank mit dem folgenden Befehl neu:

```
sudo apt-get update
```

4. Installieren der MongoDB-Shell.

```
sudo apt-get install -y mongodb-org-shell
```

Weitere Informationen zum Installieren von früheren Versionen von MongoDB auf Ihrem Ubuntu-System finden Sie unter Installieren von MongoDB Community Edition auf Ubuntu.

On other operating systems

Informationen zum Installieren der mongo-Shell auf anderen Betriebssystemen finden Sie unter Installieren von MongoDB Community Edition in der MongoDB-Dokumentation.

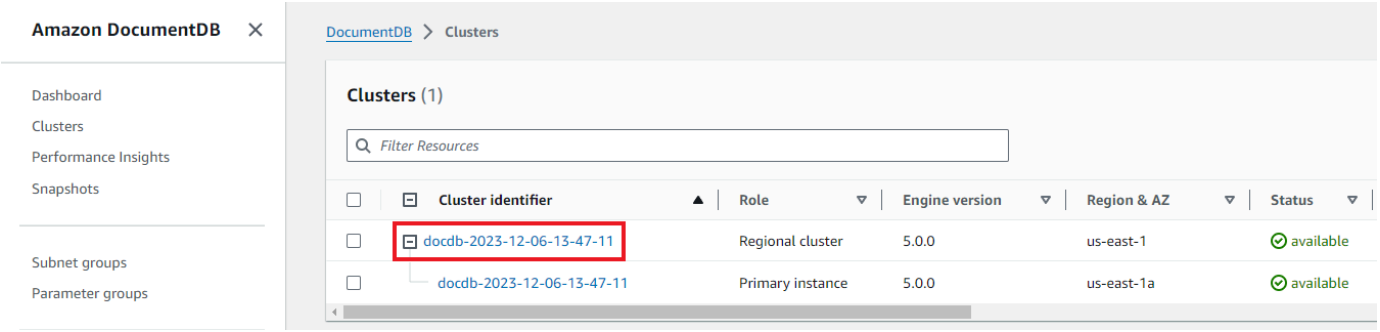
Schritt 6: Verwalten von Amazon DocumentDB TLS

Laden Sie das CA-Zertifikat für Amazon DocumentDB mit dem folgenden Code herunter: `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

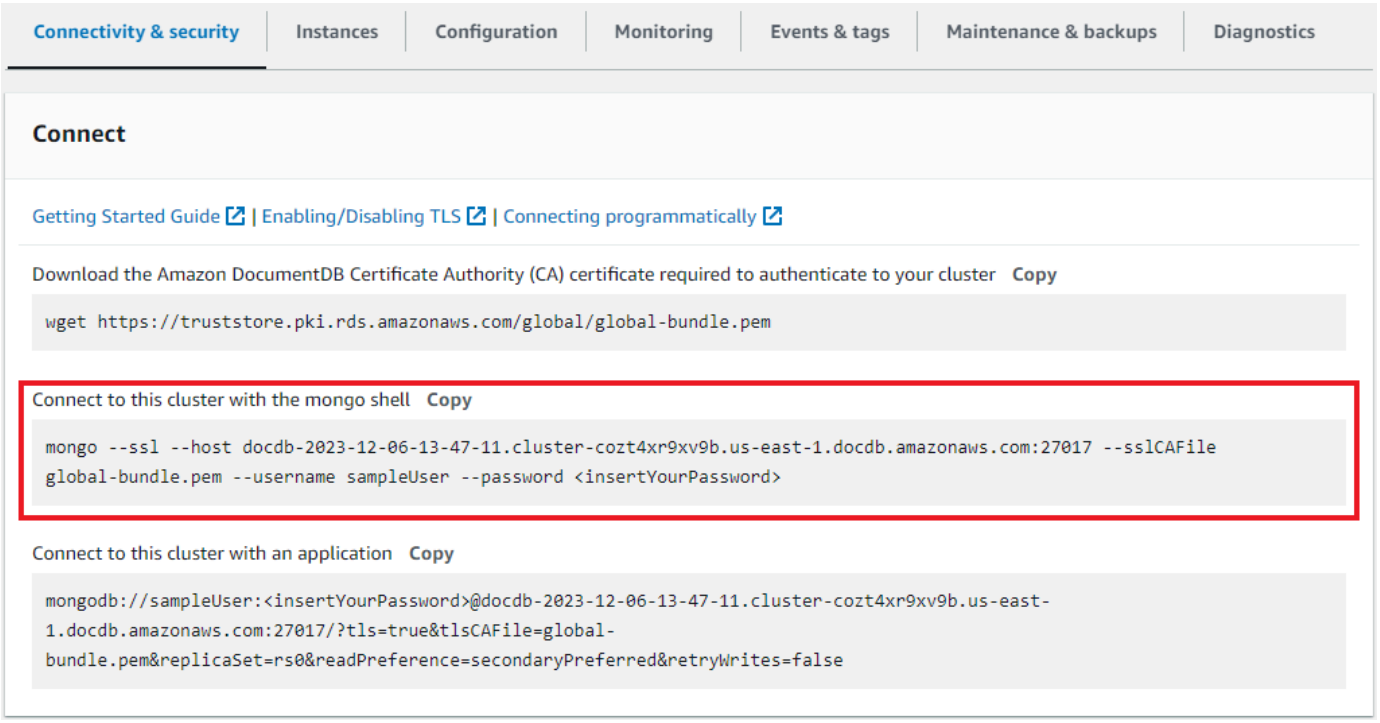
> **Note**
>
> Transport Layer Security (TLS) ist standardmäßig für alle neuen Amazon DocumentDB-Cluster aktiviert. Weitere Informationen finden Sie unter Verwalten von Amazon DocumentDB-Cluster-TLS-Einstellungen.

Schritt 7: Herstellen einer Verbindung mit Ihrem Amazon DocumentDB-Cluster

1. Suchen Sie in der Amazon DocumentDB-Konsole unter Cluster Ihren Cluster. Wählen Sie den Cluster aus, den Sie erstellt haben, indem Sie auf die Cluster-ID klicken.



2. Suchen Sie auf der Registerkarte Konnektivität und Sicherheit im Feld Verbinden die Option Verbindung zu diesem Cluster mit der mongo-Shell herstellen:



   Kopieren Sie die angegebene Verbindungszeichenfolge und fügen Sie sie in Ihr Terminal ein.

   Nehmen Sie die folgenden Änderungen vor:

   a. Stellen Sie sicher, dass Sie den richtigen Benutzernamen in der Zeichenfolge haben.

   b. Lassen Sie es weg, `<insertYourPassword>` damit Sie von der mongo-Shell beim Herstellen einer Verbindung zur Eingabe des Passworts aufgefordert werden.

Ihre Verbindungszeichenfolge sollte in etwa wie folgt aussehen:

```
mongo --ssl host docdb-2020-02-08-14-15-11.
cluster.region.docdb.amazonaws.com:27107 --sslCAFile global-bundle.pem
--username demoUser --password
```

3. Drücken Sie die Eingabetaste in Ihrem Terminal. Sie werden jetzt zur Eingabe Ihres Passworts aufgefordert. Geben Sie Ihr Passwort ein.
4. Wenn Sie Ihr Passwort eingeben und die `rs0:PRIMARY>` Eingabeaufforderung sehen können, sind Sie erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden.

Haben Sie Probleme mit der Verbindung? Weitere Informationen finden Sie unter Fehlerbehebung bei Amazon DocumentDB.

Schritt 8: Einfügen und Abfragen von Daten

Nachdem Sie nun mit Ihrem Cluster verbunden sind, können Sie einige Abfragen ausführen, um sich mit der Verwendung einer Dokumentdatenbank vertraut zu machen.

1. Um ein einzelnes Dokument einzufügen, geben Sie Folgendes ein:

```
db.collection.insert({"hello":"DocumentDB"})
```

2. Sie erhalten die folgende Ausgabe:

```
WriteResult({ "nInserted" : 1 })
```

3. Sie können das Dokument lesen, das Sie mit dem `findOne()` Befehl geschrieben haben (da es nur ein einzelnes Dokument zurückgibt). Geben Sie Folgendes ein:

```
db.collection.findOne()
```

4. Sie erhalten die folgende Ausgabe:

```
{ "_id" : ObjectId("5e401fe56056fda7321fbd67"), "hello" :
"DocumentDB" }
```

5. Um einige weitere Abfragen durchzuführen, sollten Sie einen Anwendungsfall für Spieleprofile in Betracht ziehen. Fügen Sie zunächst einige Einträge in eine Sammlung mit dem Titel ein`profiles`. Geben Sie Folgendes ein:

```
db.profiles.insertMany([
           { "_id" : 1, "name" : "Matt", "status": "active", "level": 12,
 "score":202},
           { "_id" : 2, "name" : "Frank", "status": "inactive", "level": 2,
 "score":9},
           { "_id" : 3, "name" : "Karen", "status": "active", "level": 7,
 "score":87},
           { "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
 "score":27}
           ])
```

6. Sie erhalten die folgende Ausgabe:

```
{ "acknowledged" : true, "insertedIds" : [ 1, 2, 3, 4 ] }
```

7. Verwenden Sie den `find()` Befehl , um alle Dokumente in der Profilsammlung zurückzugeben. Geben Sie Folgendes ein:

```
db.profiles.find()
```

8. Sie erhalten eine Ausgabe, die mit den Daten übereinstimmt, die Sie in Schritt 5 eingegeben haben.
9. Verwenden Sie eine Abfrage für ein einzelnes Dokument mithilfe eines Filters. Geben Sie Folgendes ein:

```
db.profiles.find({name: "Katie"})
```

10. Sie sollten diese Ausgabe zurückgeben:

```
{ "_id" : 4, "name" : "Katie", "status": "active", "level": 3,
"score":27}
```

11. Versuchen wir nun, ein Profil zu finden und es mit dem `findAndModify` Befehl zu ändern. Wir geben dem Benutzer mit dem folgenden Code zusätzliche zehn Punkte:

```
    db.profiles.findAndModify({
        query: { name: "Matt", status: "active"},
        update: { $inc: { score: 10 } }
    })
```

12. Sie erhalten die folgende Ausgabe (beachten Sie, dass sein Wert noch nicht gestiegen ist):

```
{
"_id" : 1,
"name" : "Matt",
"status" : "active",
"level" : 12,
"score" : 202
}
```

13. Mit der folgenden Abfrage können Sie überprüfen, ob sich sein Wert geändert hat:

```
db.profiles.find({name: "Matt"})
```

14. Sie erhalten die folgende Ausgabe:

```
{ "_id" : 1, "name" : "Matt", "status" : "active", "level" : 12,
"score" : 212 }
```

Schritt 9: Erkunden

Herzlichen Glückwunsch! Sie haben den Schnellstart für Amazon DocumentDB erfolgreich abgeschlossen.

Was ist als Nächstes? Erfahren Sie, wie Sie diese leistungsstarke Datenbank mit einigen ihrer beliebten Funktionen vollständig nutzen können:

- Verwalten von Amazon DocumentDB
- Skalierung
- Sichern und Wiederherstellen

> **Note**
>
> Um Kosten zu sparen, können Sie entweder Ihren Amazon DocumentDB-Cluster stoppen, um die Kosten zu senken, oder den Cluster löschen. Standardmäßig stoppt Ihre AWS Cloud9 Umgebung nach 30 Minuten Inaktivität die zugrunde liegende Amazon EC2.

Stellen Sie mithilfe des Amazon DocumentDB DocumentDB-JDBC-Treibers eine Verbindung her

Der JDBC-Treiber für Amazon DocumentDB bietet eine relationale SQL-Schnittstelle für Entwickler und ermöglicht die Konnektivität von BI-Tools wie Tableau und DbVisualizer.

Weitere Informationen finden Sie in der Amazon DocumentDB JDBC-Treiberdokumentation unter GitHub.

Themen

Erste Schritte

Schritt 1. Erstellen Sie einen Amazon DocumentDB

Wenn Sie keinen Amazon DocumentDB-Cluster erstellt haben, erstellen Sie einen anhand der Anweisungen im Abschnitt Erste Schritte im Amazon DocumentDB DocumentDB-Entwicklerhandbuch.

> **Note**
>
> DocumentDB ist ein ausschließlich Virtual Private Cloud (VPC) Wenn Sie eine Verbindung von einem lokalen Computer außerhalb der VPC des Clusters aus herstellen, müssen Sie eine SSH-Verbindung zu einer Amazon EC2 EC2-Instance herstellen. In diesem Fall starten Sie Ihren Cluster mithilfe der Anweisungen unter Connect with EC2. Weitere Informationen zum SSH-Tunneling und wann Sie es möglicherweise benötigen, finden Sie unter Verwenden eines SSH-Tunnels zur Connect mit Amazon DocumentDB.

Schritt 2. JRE- oder JDK-Installation

Abhängig von Ihrer BI-Anwendung müssen Sie möglicherweise sicherstellen, dass eine 64-Bit-JRE- oder JDK-Installationsversion 8 oder höher auf Ihrem Computer installiert ist. Sie können das Java SE Runtime Environment 8 hier herunterladen.

Schritt 3. Laden Sie den DocumentDB-JDBC-Treiber herunter

Laden Sie den DocumentDB-JDBC-Treiber von hier herunter. Der Treiber ist als einzelne JAR-Datei verpackt (z. B. documentdb-jdbc-1.0.0-all.jar).

Schritt 4. Verwenden eines SSH-Tunnels zur Connect Amazon DocumentDB

Amazon-DocumentDocumentDocumentDocumentDocumentDocumentDocumentDocumentDocumentDocumentDo Private-rivate- Auf sie kann direkt von Amazon EC2 EC2-Instances oder anderenAWS Services zugegriffen werden, die in derselben Amazon VPC bereitgestellt werden. Darüber hinaus kann über EC2a-Instances oder andereAWS Services in verschiedenen VPCs in derselbenAWS Region oder anderen Regionen über VPC-Peering auf Amazon DocumentDB zugegriffen werden.

Sie können SSH-Tunneling (auch Portweiterleitung genannt) verwenden, um von außerhalb der VPC des Clusters auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen. Dies wird bei den meisten Benutzern der Fall sein, die ihre Anwendung nicht auf einer VM in derselben VPC wie der DocumentDB-Cluster ausführen.

Um einen SSH-Tunnel zu erstellen, benötigen Sie eine Amazon EC2 Instance, die in der gleichen Amazon VPC ausgeführt wird wie Ihr Amazon DocumentDB Sie können entweder eine vorhandene EC2-Instance in derselben VPC wie Ihr Cluster verwenden oder eine erstellen. Sie können einen SSH-Tunnel zum Amazon DocumentDB-Cluster einrichten,`sample-cluster.node.us-east-1.docdb.amazonaws.com` indem Sie den folgenden Befehl auf Ihrem lokalen Computer ausführen.

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -
N
```

Das Flag -L wird für die Weiterleitung eines lokalen Ports verwendet. Dies ist eine Voraussetzung für die Verbindung mit einem BI-Tool, das auf einem Client außerhalb Ihrer VPC ausgeführt wird. Sobald Sie den obigen Schritt ausgeführt haben, können Sie mit den nächsten Schritten für das BI-Tool Ihrer Wahl fortfahren.

Weitere Informationen zum SSH-Tunneling finden Sie in der Dokumentation zur Verwendung eines SSH-Tunnels zur Verbindung mit Amazon DocumentDB.

Stellen Sie von Tableau Desktop aus eine Connect zu Amazon DocumentDB her

Themen

Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers

Um von Tableau Desktop aus eine Verbindung zu Amazon DocumentDB herzustellen, müssen Sie den DocumentDB-JDBC-Treiber und den DocumentDB Tableau-Connector herunterladen und installieren.

1. Laden Sie die DocumentDB-JDBC-Treiber-JAR-Datei herunter und kopieren Sie sie je nach Betriebssystem in eines der folgenden Verzeichnisse:
   - Fenster - `C:\Program Files\Tableau\Drivers`
   - macOS - `~/Library/Tableau/Drivers`
2. Laden Sie den DocumentDB Tableau-Connector (eine TACO-Datei) herunter und kopieren Sie ihn in Ihr Verzeichnis My Tableau Repository/Connectors.
   - Fenster - `C:\Users\[user]\Documents\My Tableau Repository\Connectors`
   - macOS - `/Users/[user]/Documents/My Tableau Repository/Connectors`

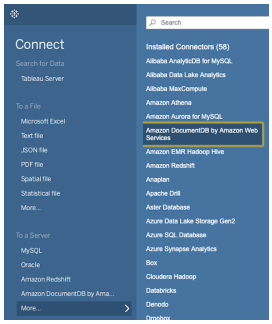
Weitere Informationen finden Sie in der Tableau-Dokumentation.

Verbindung zu Amazon DocumentDB mithilfe von Tableau herstellen - SSH Tunnel

Um von einem Client-Computer außerhalb der VPC Ihres DocumentDB-Clusters aus eine Verbindung zu Tableau herzustellen, müssen Sie einen SSH-Tunnel einrichten, bevor Sie die folgenden Schritte ausführen:

1. Starten Sie die Tableau Desktop-Anwendung.

2. Navigieren Sie zu Connect > Mit einem Server > Mehr.
3. Wählen Sie unter Installierte Connectors Amazon DocumentDB von Amazon Web Services aus.

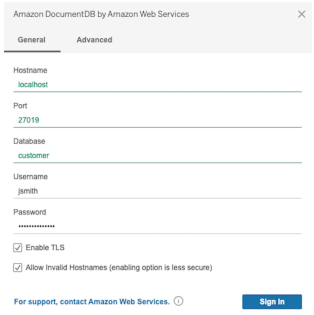


Verbindung zu Amazon DocumentDB mithilfe von Tableau herstellen — Externer SSH-Tunnel

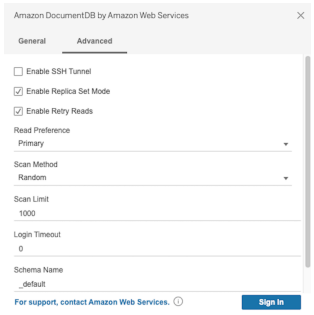
1. Geben Sie die erforderlichen Verbindungsparameter Hostname, Port, Datenbank, Benutzername und Passwort ein. Die Verbindungsparameter im folgenden Beispiel entsprechen der JDBC-Verbindungszeichenfolge:

   ```
   jdbc:documentdb://localhost:27019/test?
   tls=true&tlsAllowInvalidHostnames=true&scanMethod=random&scanLimit=1000&logir
   ```

   die Parameter Benutzername und Passwort getrennt in einer Eigenschaftensammlung übergeben werden. Weitere Informationen zu den Parametern für Verbindungszeichenfolgen finden Sie in der Github-Dokumentation zum Amazon DocumentDB-JDBC-Treiber.



2. (Optional) Weitere erweiterte Optionen finden Sie auf der Registerkarte Erweitert.

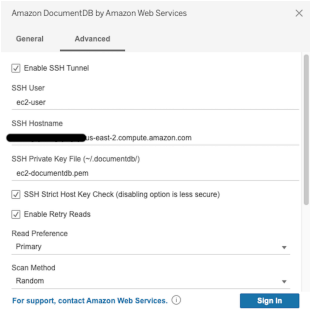


3. Klicken Sie auf Sign in.

Verbindung zu Amazon DocumentDB mithilfe von Tableau herstellen — Interner SSH-Tunnel

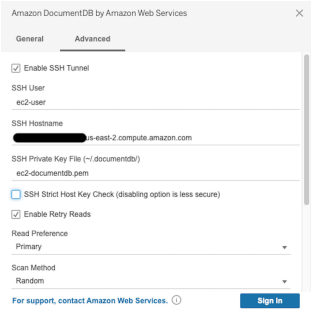
> **Note**
>
> Wenn Sie den SSH-Tunnel nicht über ein Terminal einrichten möchten, können Sie die Tableau-GUI verwenden, um Ihre EC2-Instanzdetails anzugeben, die der JDBC-Treiber standardmäßig verwendet, um einen SSH-Tunnel zu erstellen.

1. Wählen Sie auf der Registerkarte Erweitert die Option SSH-Tunnel aktivieren, um weitere Eigenschaften zu überprüfen.



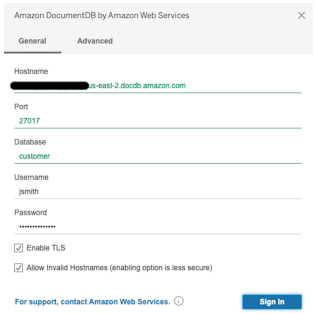
2. Geben Sie den SSH-Benutzer, den SSH-Hostnamen und die private SSH-Schlüsseldatei ein.
3. (Optional) Sie können die Option SSH Strict Host Key Check deaktivieren, wodurch die Hostschlüsselüberprüfung anhand einer bekannten Hosts-Datei umgangen wird.

> **Note**
>
> Das Deaktivieren dieser Option ist weniger sicher, da dies zu einem man-in-the-middleAngriff führen kann.



4. Geben Sie die erforderlichen Parameter ein: Hostname, Port, Datenbank, Benutzername und Passwort.

> **Note**
>
> Stellen Sie sicher, dass Sie den DocumentDB-Cluster-Endpunkt und nicht localhost verwenden, wenn Sie die interne SSH-Tunneloption verwenden.



5. Klicken Sie auf Sign In.

Connect zu Amazon DocumentDB her von DbVisualizer

Themen

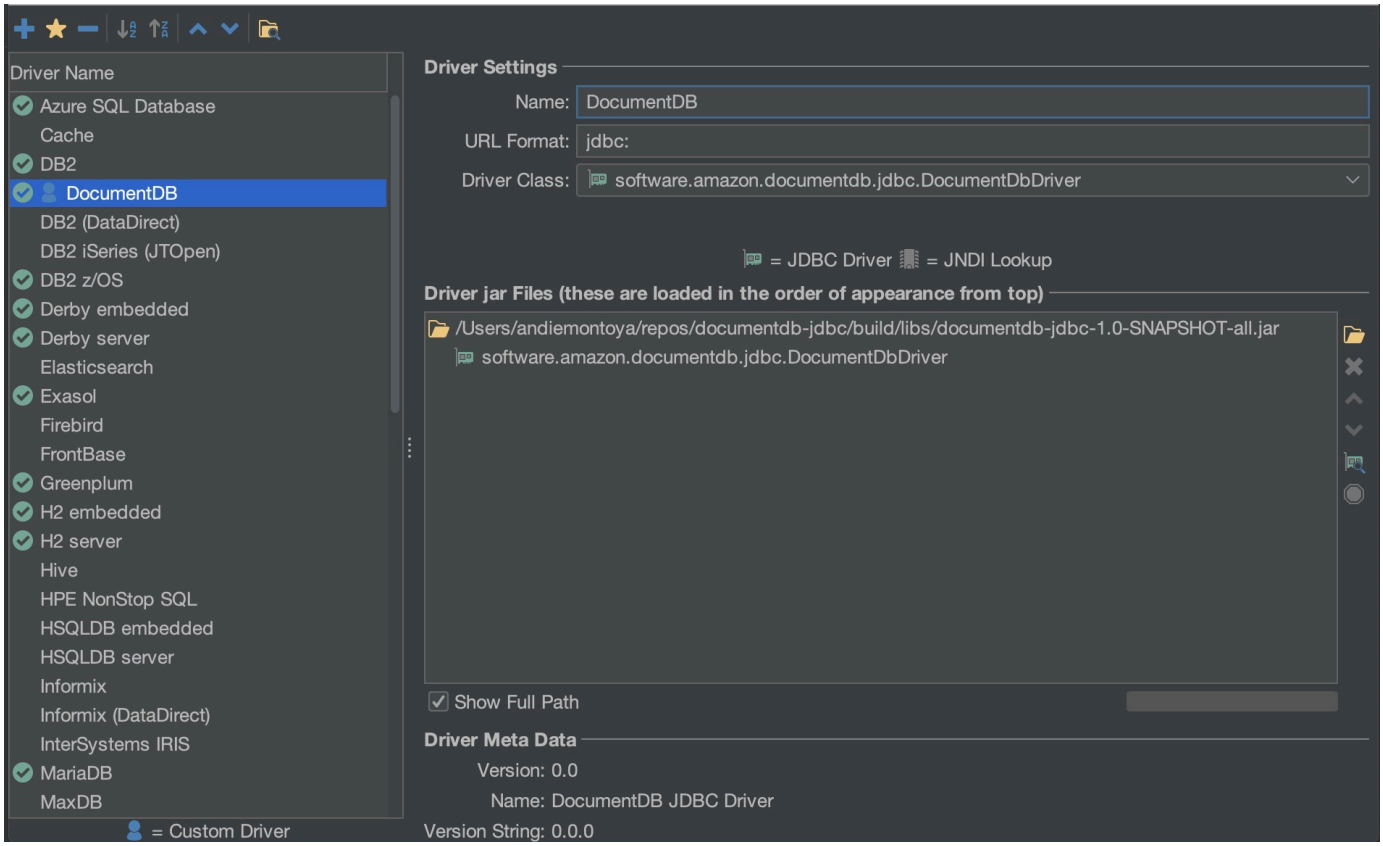
- Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers
- Verbindung zu Amazon DocumentDB herstellen mit DbVisualizer

Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers

Um eine Verbindung zu Amazon DocumentDB herzustellen, müssen DbVisualizer Sie zuerst den Amazon DocumentDB DocumentDB-JDBC-Treiber importieren.

1. Starten Sie die DbVisualizer Anwendung und navigieren Sie zum Menüpfad: Tools > Driver Manager...
2. Wählen Sie + (oder wählen Sie im Menü Treiber > Treiber erstellen).
3. Legen Sie Name auf `DocumentDB` fest.
4. Stellen Sie das URL-Format auf`jdbc:documentdb://<host>[:port]/<database>[?option=value[&option=value[...]]]`

5. Wählen Sie die Ordnerschaltfläche und dann die Amazon DocumentDB DocumentDB-JDBC-Treiber-JAR-Datei aus und klicken Sie auf die Schaltfläche Öffnen.
6. Vergewissern Sie sich, dass das Feld Fahrerklasse auf gesetzt ist`software.amazon.documentdb.jdbc.DocumentDbDriver`. Ihre Driver Manager-Einstellungen für DocumentDB sollten wie im folgenden Beispiel aussehen.



7. Schließen Sie das Dialogfeld. Der Amazon DocumentDB DocumentDB-JDBC-Treiber wird eingerichtet und kann verwendet werden.

Verbindung zu Amazon DocumentDB herstellen mit DbVisualizer

Connect zu Amazon DocumentDB her über DbVisualizer

1. Wenn Sie eine Verbindung von außerhalb der VPC des Amazon DocumentDB-Clusters herstellen, stellen Sie sicher, dass Sie einen SSH-Tunnel eingerichtet haben.
2. Wählen Sie im Menü der obersten Ebene Datenbank > Datenbankverbindung erstellen.
3. Geben Sie einen aussagekräftigen Namen für das Feld Name ein.
4. Stellen Sie Driver (JDBC) auf den DocumentDB-Treiber ein, den Sie im vorherigen Abschnitt erstellt haben.

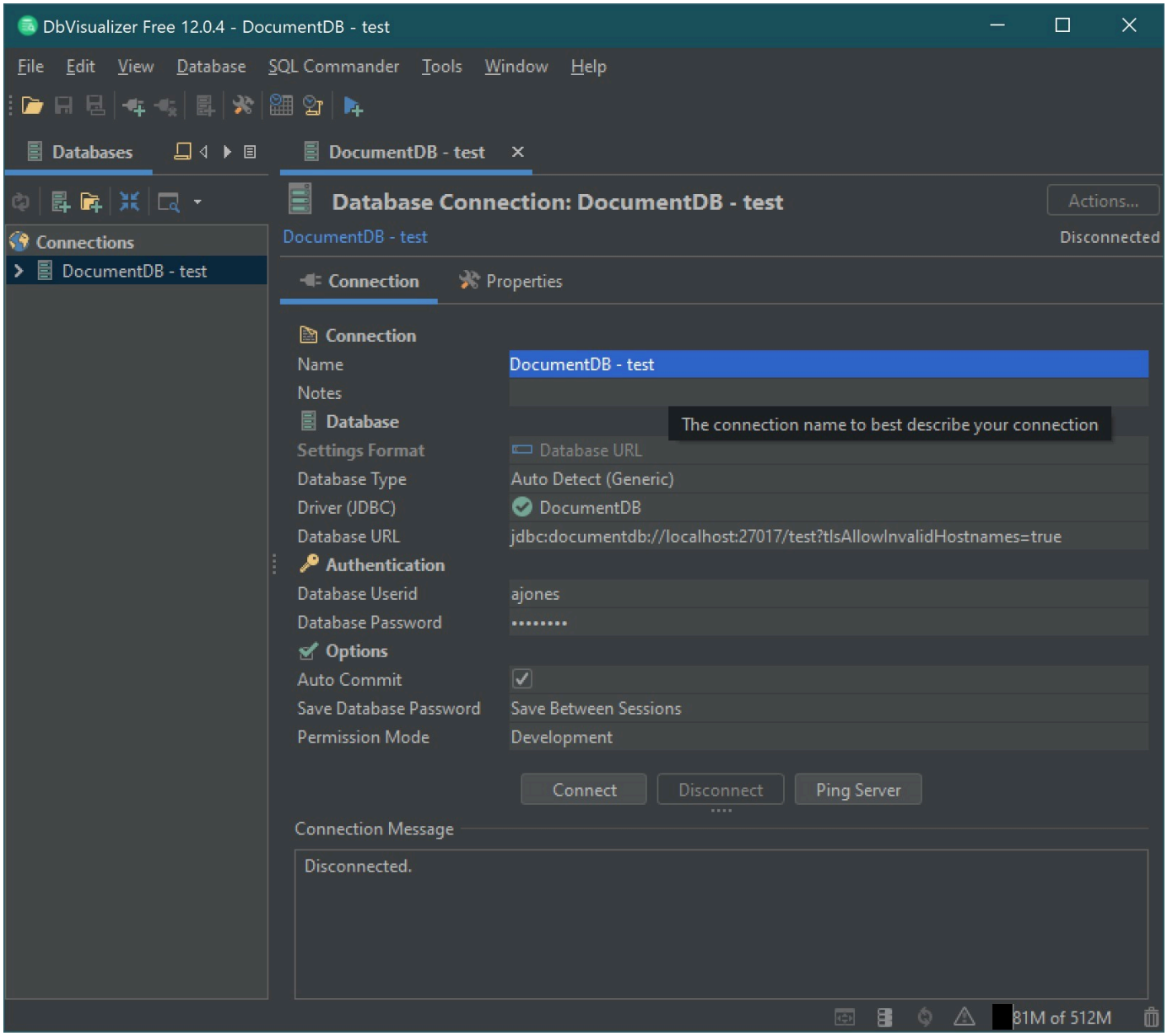
5. Stellen Sie die Datenbank-URL auf Ihre JDBC-Verbindungszeichenfolge ein.

   Beispiel: `jdbc:documentdb://localhost:27017/database?tlsAllowInvalidHostnames=true`

6. Stellen Sie die Datenbankbenutzer-ID auf Ihre Amazon DocumentDB DocumentDB-Benutzer-ID ein.

7. Stellen Sie das Datenbankkennwort auf das entsprechende Passwort für die Benutzer-ID ein.

   Ihr Datenbankverbindungsdialogfeld sollte wie im folgenden aussehen:



8. Wählen Sie Connect (Verbinden) aus.

Automatische JDBC-Schemagenerierung

Amazon DocumentDB ist eine Dokumentendatenbank und hat daher nicht das Konzept von Tabellen und Schema. BI-Tools wie Tableau erwarten jedoch, dass die Datenbank, mit der sie verbunden sind, ein Schema enthält. Insbesondere wenn die JDBC-Treiberverbindung das Schema für die Sammlung in der Datenbank abrufen muss, fragt sie nach allen Sammlungen in der Datenbank ab. Der Treiber ermittelt, ob eine zwischengespeicherte Version des Schemas für diese Sammlung bereits existiert. Wenn keine zwischengespeicherte Version vorhanden ist, durchsucht sie die Sammlung nach Dokumenten und erstellt ein Schema, das auf dem folgenden Verhalten basiert.

Themen

Einschränkungen der Schemagenerierung

Der DocumentDB-JDBC-Treiber begrenzt die Länge der Identifikatoren auf 128 Zeichen. Der Schemagenerator kann die Länge der generierten Bezeichner (Tabellennamen und Spaltennamen) kürzen, um sicherzustellen, dass sie dieser Grenze entsprechen.

Optionen für die Scanmethode

Das Sampling-Verhalten kann mithilfe von Verbindungszeichenfolgen oder Datenquellenoptionen geändert werden.

- Scan-Methode = <option>
  - random - (Standard) - Die Beispieldokumente werden in zufälliger Reihenfolge zurückgegeben.
  - idForward - Die Beispieldokumente werden in der Reihenfolge ihrer ID zurückgegeben.
  - idReverse — Die Beispieldokumente werden in umgekehrter Reihenfolge der ID zurückgegeben.
  - all — Wählen Sie alle Dokumente in der Sammlung aus.
- scanLimit= <n>- Die Anzahl der zu scannenden Dokumente. Der Wert muss eine positive ganze Zahl sein. Der Standardwert lautet 1 000. Wenn scanMethod auf all gesetzt ist, wird diese Option ignoriert.

Amazon DocumentDB DocumententDocumentDocument

Der DocumentDB-Server unterstützt eine Reihe von MongoDB-Datentypen. Nachfolgend sind die unterstützten Datentypen und die zugehörigen JDBC-Datentypen aufgeführt.

MongoDB — Datentyp	Unterstützt in DocumentDB	JDBC-Datentyp
Binäre Daten	Ja	VARBINARY
Boolesch	Ja	BOOLEAN
Double	Ja	DOUBLE
32-Bit Integer	Ja	INTEGER
64-Bit-Ganzzahl	Ja	BIGINT
Zeichenfolge	Ja	VARCHAR
ObjectId	Ja	VARCHAR
Datum	Ja	TIMESTAMP
Null	Ja	VARCHAR
Regulärer Ausdruck	Ja	VARCHAR
Zeitstempel	Ja	VARCHAR
MinKey	Ja	VARCHAR
MaxKey	Ja	VARCHAR
Objekt	Ja	virtueller Tisch
Array	Ja	virtueller Tisch
Decimal128	Nein	DECIMAL
JavaScript	Nein	VARCHAR

MongoDB — Datentyp	Unterstützt in DocumentDB	JDBC-Datentyp
JavaScript (mit Geltungsbereich)	Nein	VARCHAR
Undefined	Nein	VARCHAR
Symbol	Nein	VARCHAR
DBZeiger (4.0+)	Nein	VARCHAR

Zuordnen von skalaren Dokumentfeldern

Beim Scannen einer Stichprobe von Dokumenten aus einer Sammlung erstellt der JDBC-Treiber ein oder mehrere Schemas, um die Beispiele in der Sammlung darzustellen. Im Allgemeinen wird ein Skalarfeld im Dokument einer Spalte im Tabellenschema zugeordnet. In einer Sammlung mit dem Namen team und einem einzelnen Dokument würde dies`{ "_id" : "112233", "name" : "Alastair", "age": 25 }` beispielsweise dem Schema zugeordnet:

Tabellenname	Spaltenname	Datentyp	Schlüssel
Mannschaft	Team-ID	VARCHAR	PK
Mannschaft	Name	VARCHAR	
Mannschaft	Alter	INTEGER	

Datentyp Konfliktförderung

Beim Scannen der Stichprobendokumente ist es möglich, dass die Datentypen für ein Feld von Dokument zu Dokument nicht konsistent sind. In diesem Fall erhöht der JDBC-Treiber den JDBC-Datentyp auf einen gemeinsamen Datentyp, der für alle Datentypen aus den gesampelten Dokumenten geeignet ist.

Beispiel:

```
{
"_id" : "112233",
```

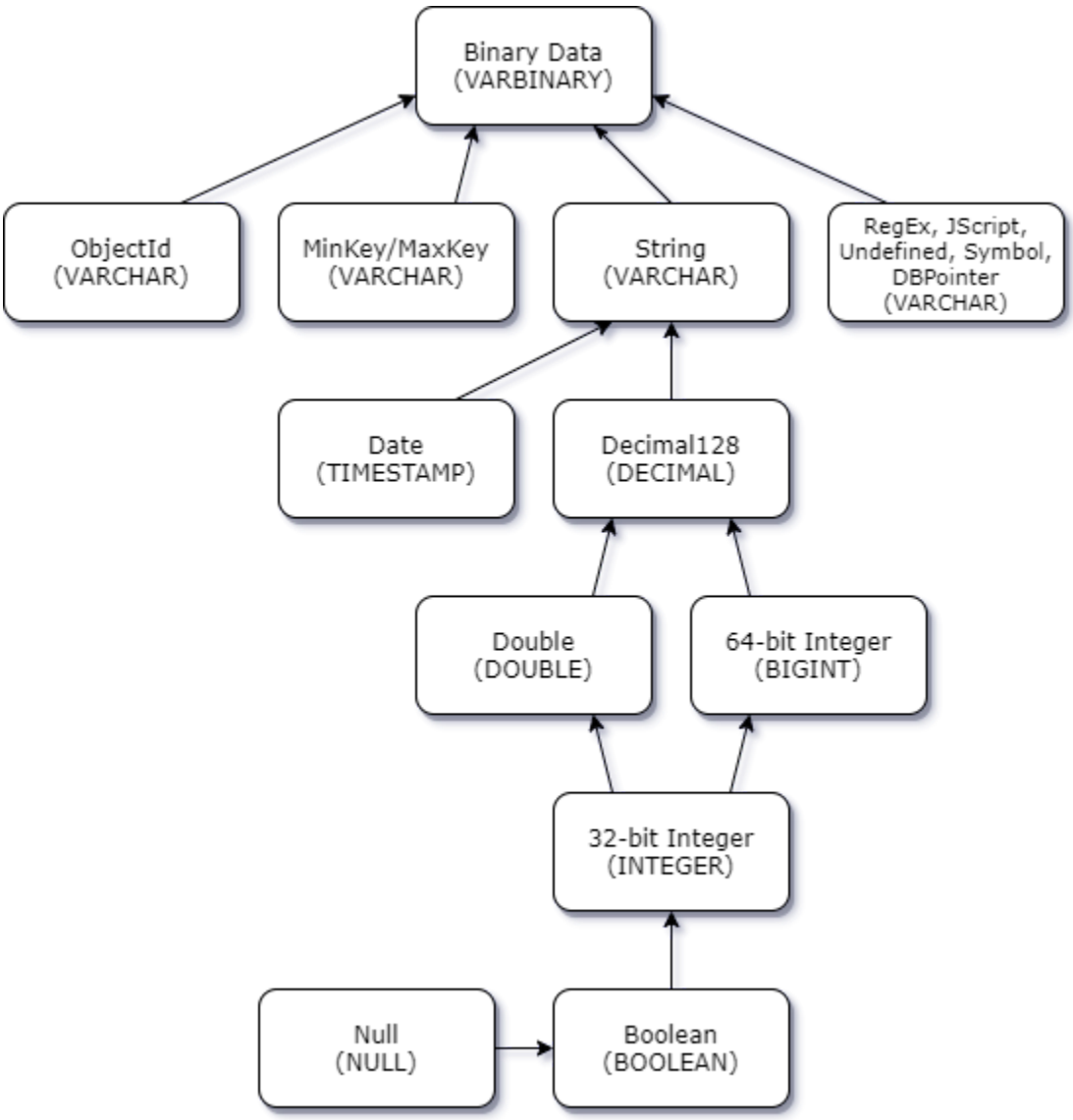
```
"name" : "Alastair", "age" : 25
}

{
"_id" : "112244",
"name" : "Benjamin",
"age" : "32"
}
```

Das Altersfeld ist im ersten Dokument vom Typ 32-Bit-Ganzzahl, im zweiten Dokument jedoch vom Typ Zeichenfolge. Hier wird der JDBC-Treiber den JDBC-Datentyp auf VARCHAR hochstufen, um einen der beiden Datentypen zu verarbeiten, wenn er auftritt.

Tabellenname	Spaltenname	Datentyp	Schlüssel
Mannschaft	Team-ID	VARCHAR	PK
Mannschaft	Name	VARCHAR	
Mannschaft	Alter	VARCHAR	

Skalarskalare Konfliktförderung

Das folgende Diagramm zeigt, wie skalar-skalare Datentypkonflikte gelöst werden.



Konfliktförderung skalarkomplexer Typen

Wie bei den skalaren Typkonflikten kann dasselbe Feld in verschiedenen Dokumenten widersprüchliche Datentypen zwischen komplex (Array und Objekt) und skalar (Ganzzahl, boolescher Wert usw.) aufweisen. Alle diese Konflikte werden für diese Felder in VARCHAR gelöst (heraufgestuft). In diesem Fall werden Array- und Objektdaten als JSON-Repräsentation zurückgegeben.

Beispiel für einen Konflikt mit einem eingebetteten Array — Zeichenkettenfeld:

```
{
   "_id":"112233",
   "name":"George Jackson",
```

```
   "subscriptions":[
      "Vogue",
      "People",
      "USA Today"
   ]
}
{
   "_id":"112244",
   "name":"Joan Starr",
   "subscriptions":1
}
```

Das obige Beispiel ist dem Schema für die Tabelle customer2 zugeordnet:

Tabellenname	Spaltenname	Datentyp	Schlüssel
Kunde 2	Kunden2-ID	VARCHAR	PK
Kunde 2	Name	VARCHAR	
Kunde 2	Abonnement	VARCHAR	

und die virtuelle Tabelle customer1_subscriptions:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer1_abonnements	Kundennummer 1	VARCHAR	PK/FK
customer1_abonnements	subscriptions_index_lvl0	BIGINT	PK
customer1_abonnements	Wert	VARCHAR	
customer_address	city	VARCHAR	
customer_address	region	VARCHAR	
customer_address	country	VARCHAR	

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer_address	Code	VARCHAR	

Behandlung von Objekt- und Array-Datentypen

Bisher haben wir nur beschrieben, wie skalare Datentypen abgebildet werden. Objekt- und Array-Datentypen werden (derzeit) virtuellen Tabellen zugeordnet. Der JDBC-Treiber erstellt eine virtuelle Tabelle, um entweder Objekt- oder Array-Felder in einem Dokument darzustellen. Der Name der zugeordneten virtuellen Tabelle verkettet den Namen der ursprünglichen Sammlung, gefolgt vom Namen des Felds, getrennt durch einen Unterstrich („_“).

Der Primärschlüssel („_id“) der Basistabelle erhält in der neuen virtuellen Tabelle einen neuen Namen und wird als Fremdschlüssel für die zugehörige Basistabelle bereitgestellt.

Für eingebettete Felder vom Typ Array werden Indexspalten generiert, um den Index im Array auf jeder Ebene des Arrays darzustellen.

Beispiel für ein eingebettetes Objektfeld

Für Objektfelder in einem Dokument wird vom JDBC-Treiber eine Zuordnung zu einer virtuellen Tabelle erstellt.

```
{
   "Collection: customer",
   "_id":"112233",
   "name":"George Jackson",
   "address":{
      "address1":"123 Avenue Way",
      "address2":"Apt. 5",
      "city":"Hollywood",
      "region":"California",
      "country":"USA",
      "code":"90210"
   }
}
```

Das obige Beispiel ist dem Schema für die Kundentabelle zugeordnet:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer	Kunden-ID	VARCHAR	PK
customer	Name	VARCHAR	

und die virtuelle Tabelle customer_address:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer_address	Kunden-ID	VARCHAR	PK/FK
customer_address	adresse1	VARCHAR	
customer_address	adresse2	VARCHAR	
customer_address	city	VARCHAR	
customer_address	region	VARCHAR	
customer_address	country	VARCHAR	
customer_address	Code	VARCHAR	

Beispiel für ein eingebettetes Array-Feld

Für Array-Felder in einem Dokument wird vom JDBC-Treiber auch eine Zuordnung zu einer virtuellen Tabelle erstellt.

```
{
   "Collection: customer1",
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
      "Vogue",
      "People",
      "USA Today"
   ]
}
```

Das obige Beispiel ist dem Schema für die Tabelle customer1 zugeordnet:

Tabellenname	Spaltenname	Datentyp	Schlüssel
Kunde 1	Kundennummer 1	VARCHAR	PK
Kunde 1	Name	VARCHAR	

und die virtuelle Tabelle customer1_subscriptions:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer1_abonnements	Kundennummer 1	VARCHAR	PK/FK
customer1_abonnements	subscriptions_index_lvl0	BIGINT	PK
customer1_abonnements	Wert	VARCHAR	
customer_address	city	VARCHAR	
customer_address	region	VARCHAR	
customer_address	country	VARCHAR	
customer_address	Code	VARCHAR	

SQL-Unterstützung und Einschränkungen

Der Amazon DocumentDB DocumentDB-JDBC-Treiber ist ein schreibgeschützter Treiber, der eine Teilmenge von SQL-92 und einige gängige Erweiterungen unterstützt. Weitere Informationen finden Sie in der Dokumentation zu den SQL-Einschränkungen und der Dokumentation zu den JDBC-Einschränkungen.

Fehlerbehebung

Wenn Sie Probleme bei der Verwendung des Amazon DocumentDB-JDBC-Treibers haben, lesen Sie den Leitfaden zur Fehlerbehebung.

Stellen Sie mithilfe des Amazon DocumentDB-ODBC-Treibers eine Verbindung her

Der ODBC-Treiber für Amazon DocumentDB bietet eine relationale SQL-Schnittstelle für Entwickler und ermöglicht die Konnektivität von BI-Tools wie Power BI Desktop und Microsoft Excel.

Weitere Informationen finden Sie in der Amazon DocumentDB ODBC-Treiberdokumentation unter GitHub.

Themen

- Erste Schritte
- Einrichten des Amazon DocumentDB-ODBC-Treibers in Windows
- Stellen Sie von Microsoft Excel aus Connect zu Amazon DocumentDB her
- Stellen Sie von Microsoft Power BI Desktop aus eine Connect zu Amazon DocumentDB her
- Automatische Schemagenerierung
- SQL-Unterstützung und Einschränkungen
- Fehlerbehebung

Erste Schritte

Schritt 1. Amazon DocumentDB-Cluster erstellen

Wenn Sie noch keinen Amazon DocumentDB Documententententententententental-Private-Cluster haben, gibt es eine Reihe von Möglichkeiten für den Einstieg.

> **Note**
>
> Amazon DocumentDB Documententententententabor (VPC). Wenn Sie von einem lokalen Computer außerhalb der VPC des Clusters aus eine Verbindung herstellen, müssen Sie eine SSH-Verbindung zu einer Amazon EC2 EC2-Instance herstellen. In diesem Fall starten Sie Ihren Cluster mithilfe der Anweisungen unter Connect with EC2. Weitere

Informationen zum SSH-Tunneling und wann Sie es möglicherweise benötigen, finden Sie unter Verwenden eines SSH-Tunnels zur Connect mit Amazon DocumentDB.

Schritt 2. JRE- oder JDK-Installation

Abhängig von Ihrer BI-Anwendung müssen Sie möglicherweise sicherstellen, dass eine 64-Bit-JRE- oder JDK-Installationsversion 8 oder höher auf Ihrem Computer installiert ist. Sie können das Java SE Runtime Environment 8 hier herunterladen.

Schritt 3. Laden Sie den Amazon DocumentDB DocumentDB-ODBC-Treiber herunter

Laden Sie den Amazon DocumentDB DocumentDB-ODBC-Treiber hier herunter. Wählen Sie das richtige Installationsprogramm (z. B. documentdb-odbc-1.0.0.msi). Folgen Sie der Installationsanleitung.

Schritt 4. Verwenden eines SSH-Tunnels zur Connect Amazon DocumentDB

Amazon-Virtual-Private-Cloud (Amazon VPC) werden in einer Amazon-Virtual-Private-Cloud (Amazon VPC) bereitgestellt. Auf sie kann direkt von Amazon EC2 EC2-Instances oder anderenAWS Services zugegriffen werden, die in derselben Amazon VPC bereitgestellt werden. Darüber hinaus kann Amazon EC2 EC2-Instances oder andereAWS Services in verschiedenen VPCs in derselbenAWS Region oder anderen Regionen über VPC-Peering auf Amazon DocumentDB zugreifen.

Nehmen wir jedoch an, dass Ihr Anwendungsfall erfordert, dass Sie (oder Ihre Anwendung) von außerhalb der VPC des Clusters auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zugreifen. Dies wird bei den meisten Benutzern der Fall sein, die ihre Anwendung nicht auf einer VM in derselben VPC wie der Amazon DocumentDB-Cluster ausführen. Wenn Sie eine Verbindung von außerhalb der VPC herstellen, können Sie SSH-Tunneling (auch Portweiterleitung genannt) verwenden, um auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen.

So erstellen Sie einen SSH-Tunnel, um einen SSH-Tunnel zu erstellen, benötigen Sie eine Amazon EC2 Instance, die in der gleichen Amazon VPC ausgeführt wird wie Ihr Amazon DocumentDB Documententententententententunnel. Sie können entweder eine vorhandene EC2-Instance in derselben VPC wie Ihr Cluster verwenden oder eine erstellen. Sie können einen SSH-Tunnel zum Amazon DocumentDB-Cluster einrichten,`sample-cluster.node.us-east-1.docdb.amazonaws.com` indem Sie den folgenden Befehl auf Ihrem lokalen Computer ausführen:

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -
N
```

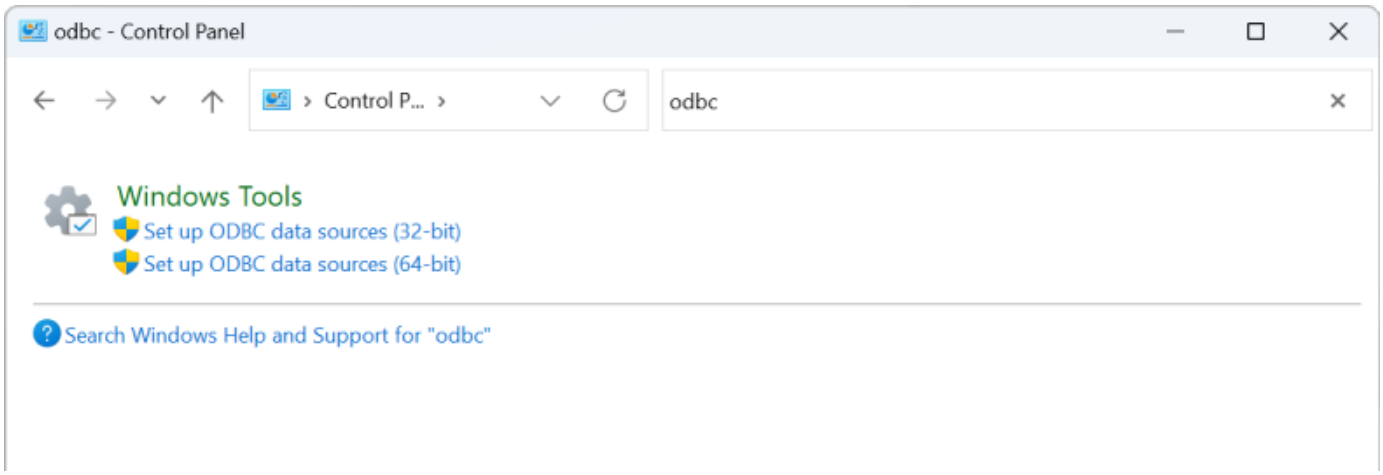
Das `-L`-Flag dient zur Weiterleitung eines lokalen Ports. Dies ist eine Voraussetzung für die Verbindung mit einem BI-Tool, das auf einem Client außerhalb Ihrer VPC ausgeführt wird. Sobald Sie den obigen Schritt ausgeführt haben, können Sie mit den nächsten Schritten für das BI-Tool Ihrer Wahl fortfahren.

Weitere Informationen zum SSH-Tunneling finden Sie in der Dokumentation zur Verwendung eines SSH-Tunnels zur Connect Amazon DocumentDB.

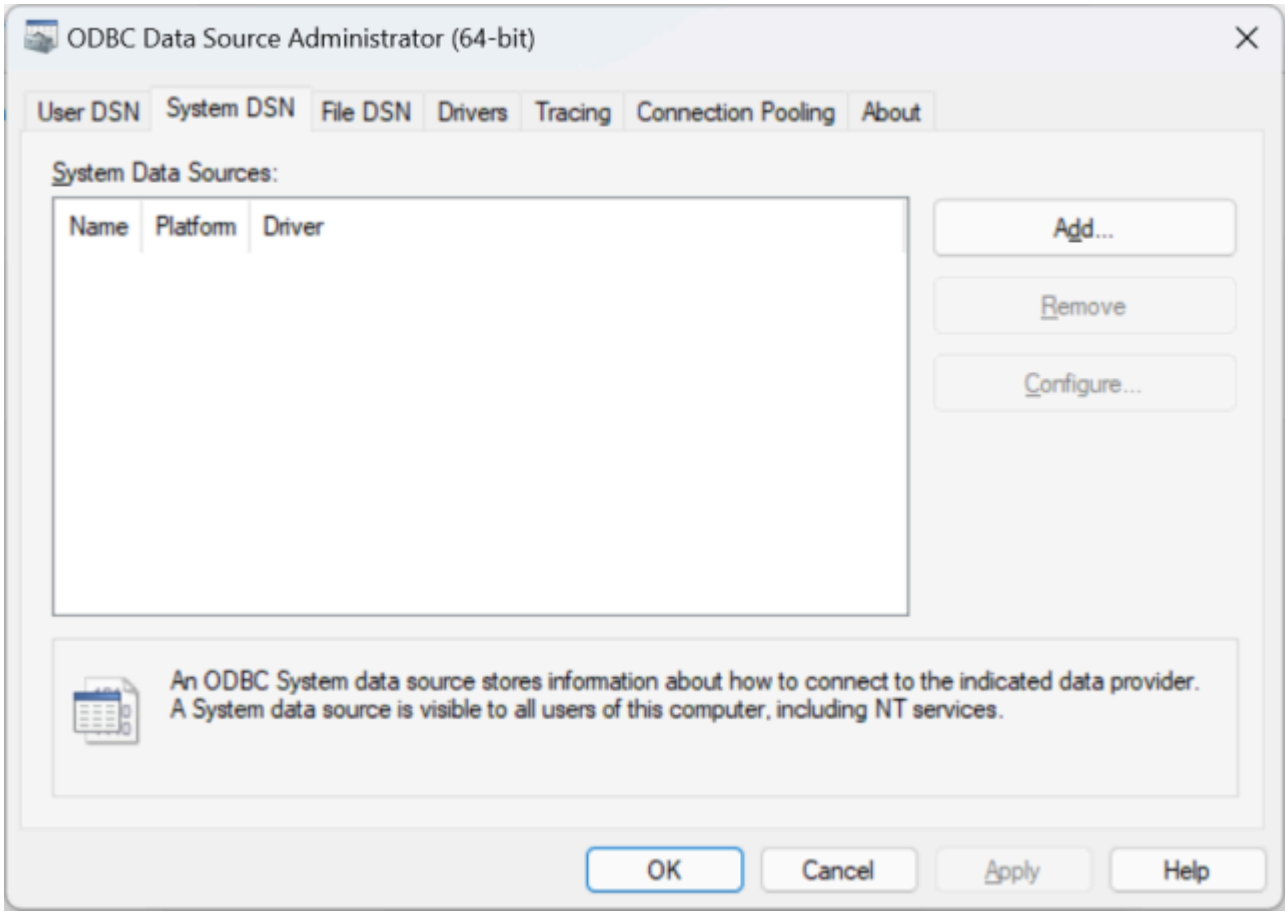
Einrichten des Amazon DocumentDB-ODBC-Treibers in Windows

Gehen Sie wie folgt vor, um den Amazon DocumentDB Documententententententententententabor in Windows einzurichten:

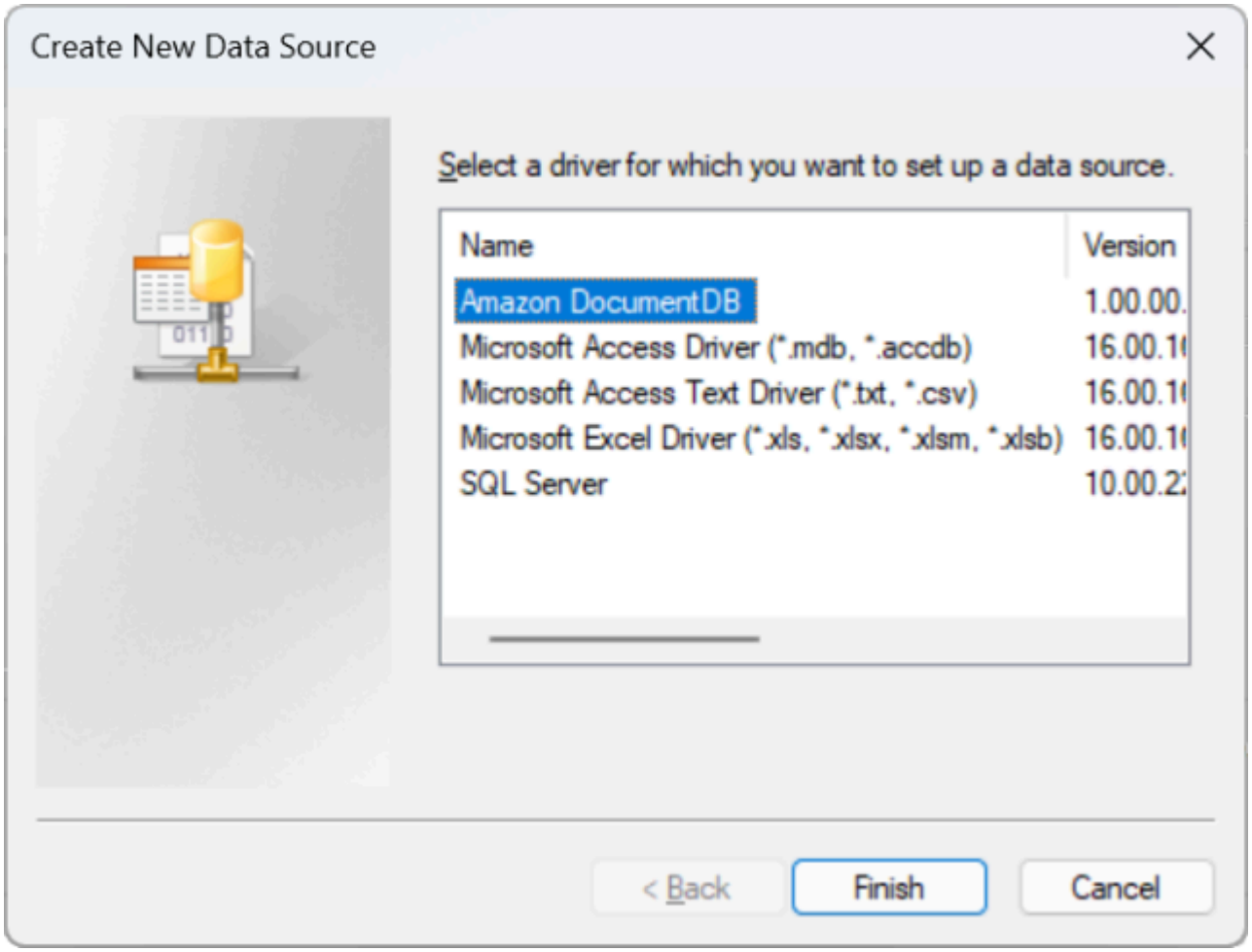
1. Öffnen Sie die Systemsteuerung in Windows und suchen Sie nach ODBC (oder wählen Sie im Menü Windows Tools > ODBC-Datenquellen (32-Bit) oder ODBC-Datenquellen (64-Bit)):



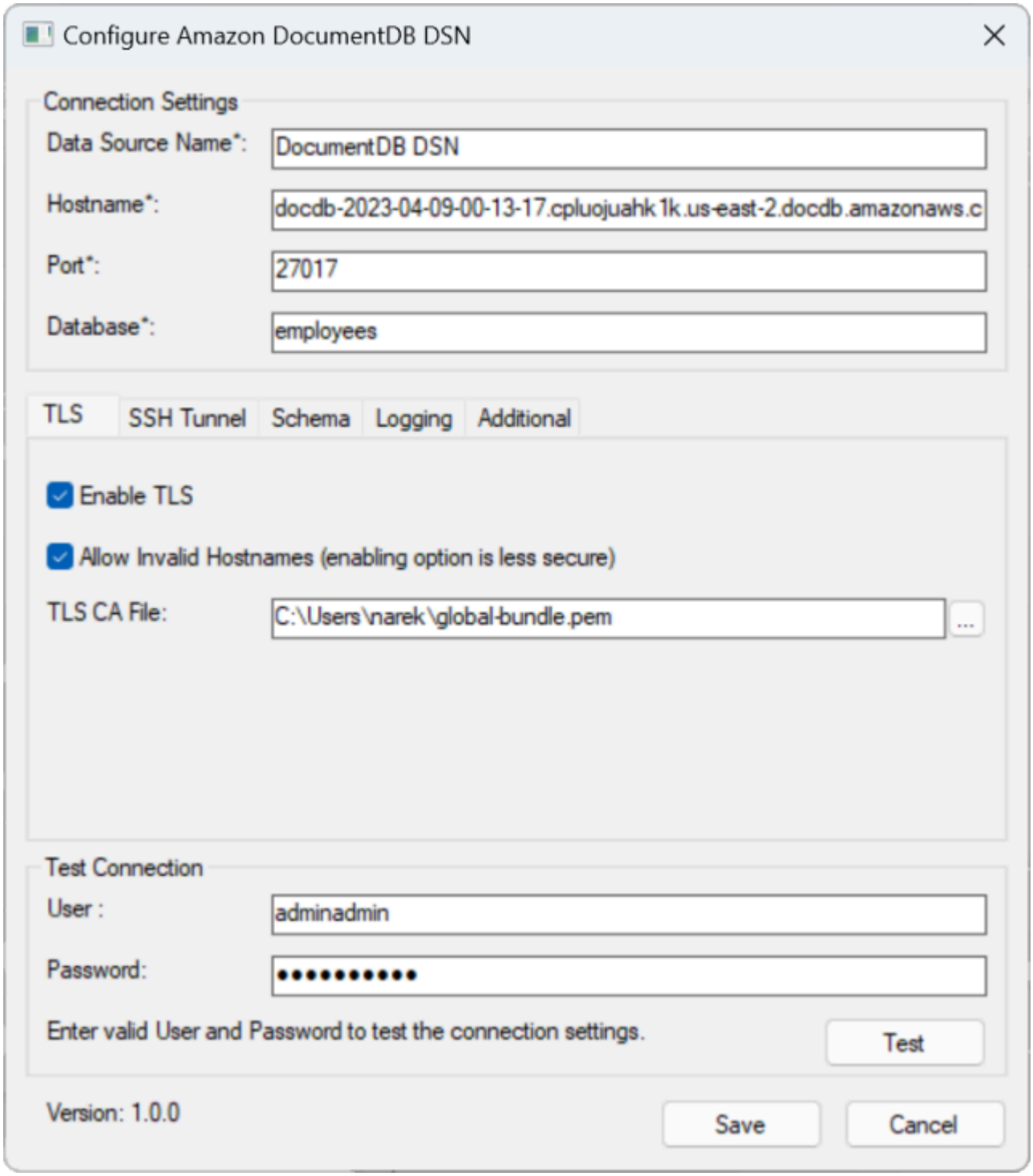
2. Wählen Sie den entsprechenden ODBC-Treiber-Datenquellenadministrator aus: Entscheiden Sie sich für die 32-Bit-Version, falls sie installiert ist, andernfalls wählen Sie die 64-Bit-Version.
3. Wählen Sie die Registerkarte System DSN und klicken Sie dann auf Hinzufügen... so fügen Sie einen neuen DSN hinzu:



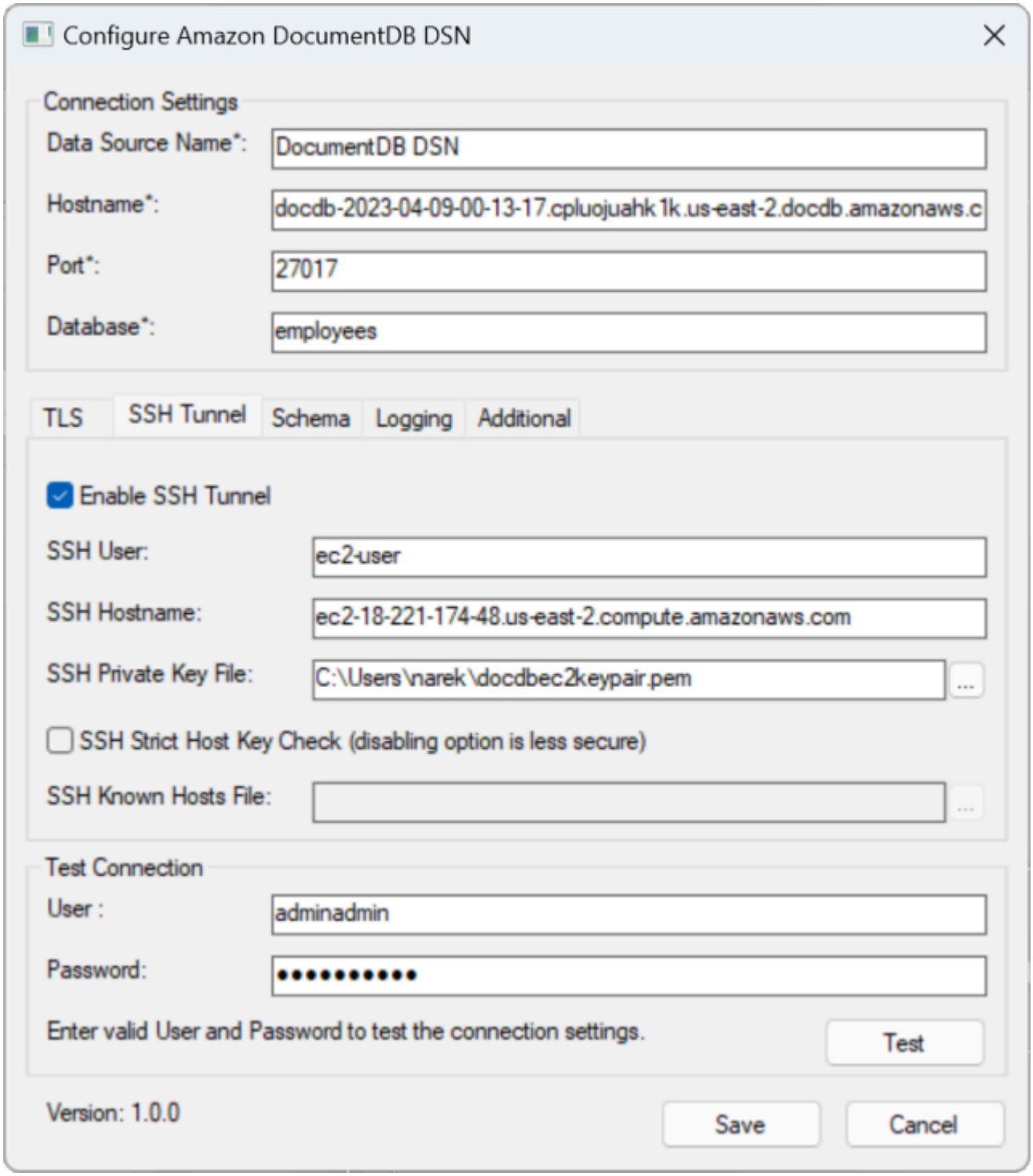
4. Wählen Sie Amazon DocumentDB aus der Liste der Datenquellentreiber aus:



5. Füllen Sie im Dialogfeld „Amazon DocumentDB DSN konfigurieren“ die Felder „Konfigurationseinstellungen“, „TLS“ und „Verbindung testen“ aus und klicken Sie dann auf Speichern:



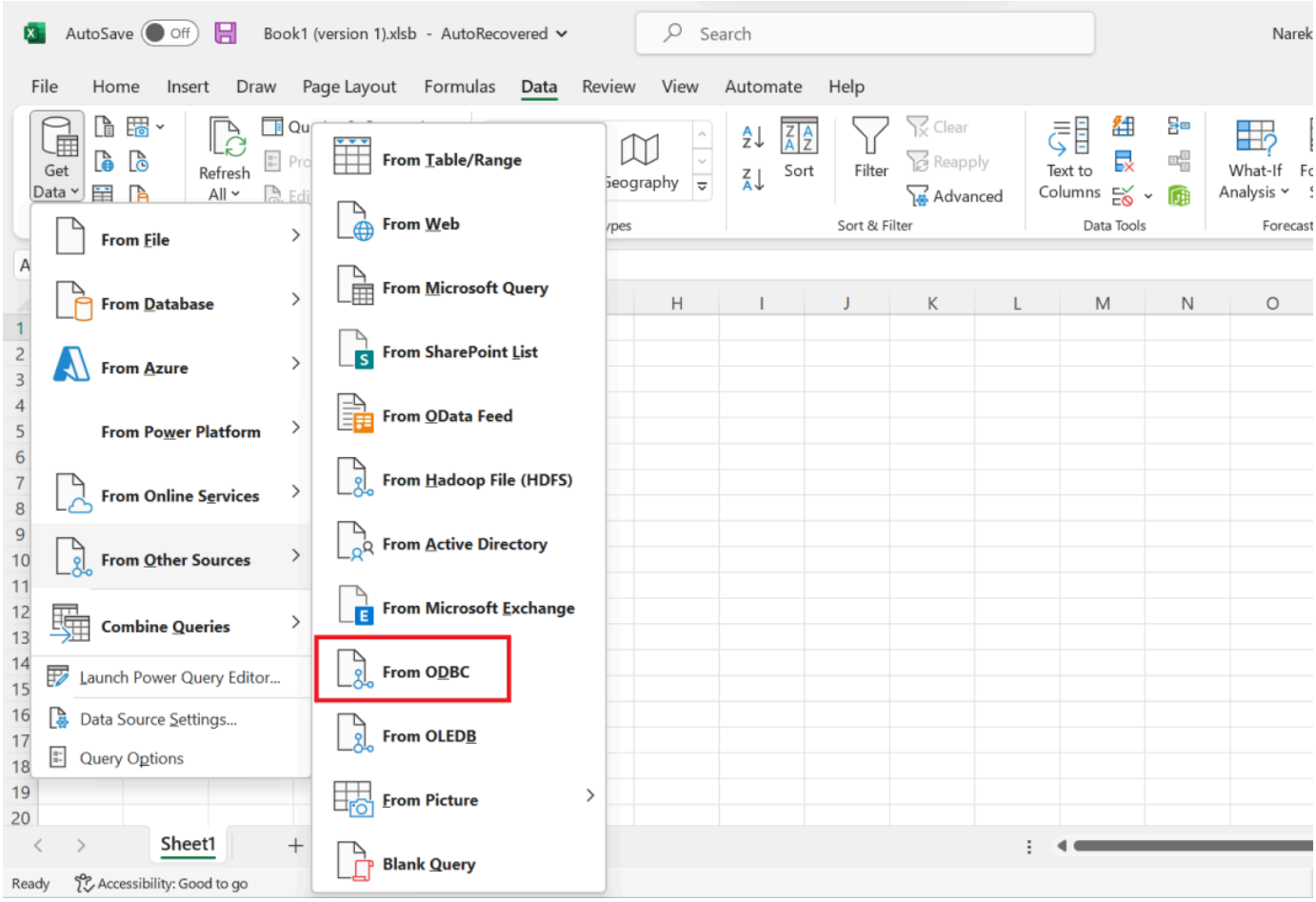
6. Stellen Sie sicher, dass Sie das Windows-Formular korrekt ausfüllen, da die Verbindungsdetails je nach der von Ihnen gewählten SSH-Tunnelmethode zur EC2-Instance unterschiedlich sein können. SSH-Tunneling-Methoden finden Sie hier. Weitere Informationen zu den einzelnen Eigenschaften finden Sie unter Syntax und Optionen der Verbindungszeichenfolge.



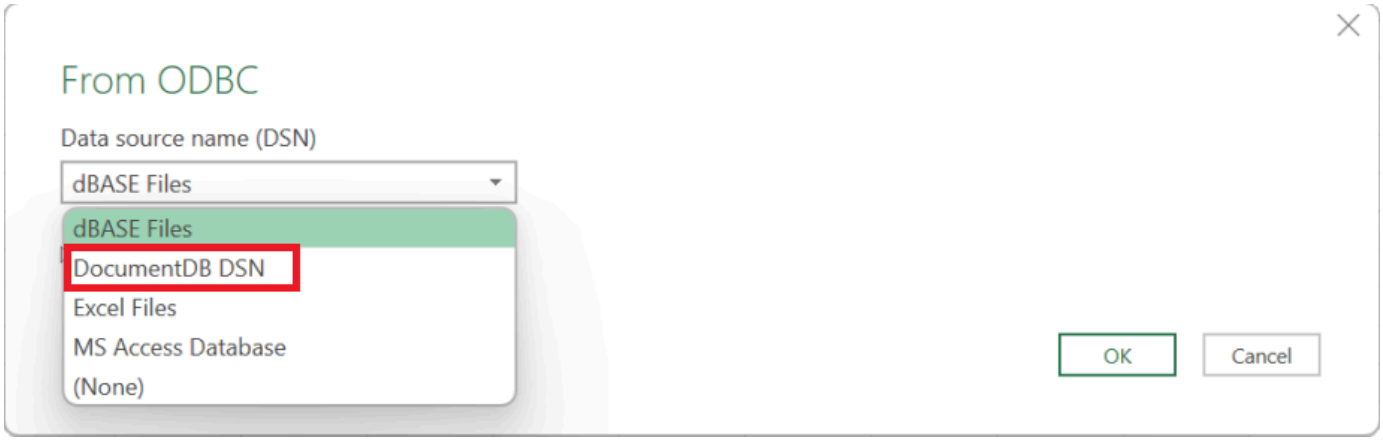
Weitere Informationen zur Konfiguration des Amazon DocumentDB-ODBC-Treibers unter Windows finden Sie hier.

Stellen Sie von Microsoft Excel aus Connect zu Amazon DocumentDB her

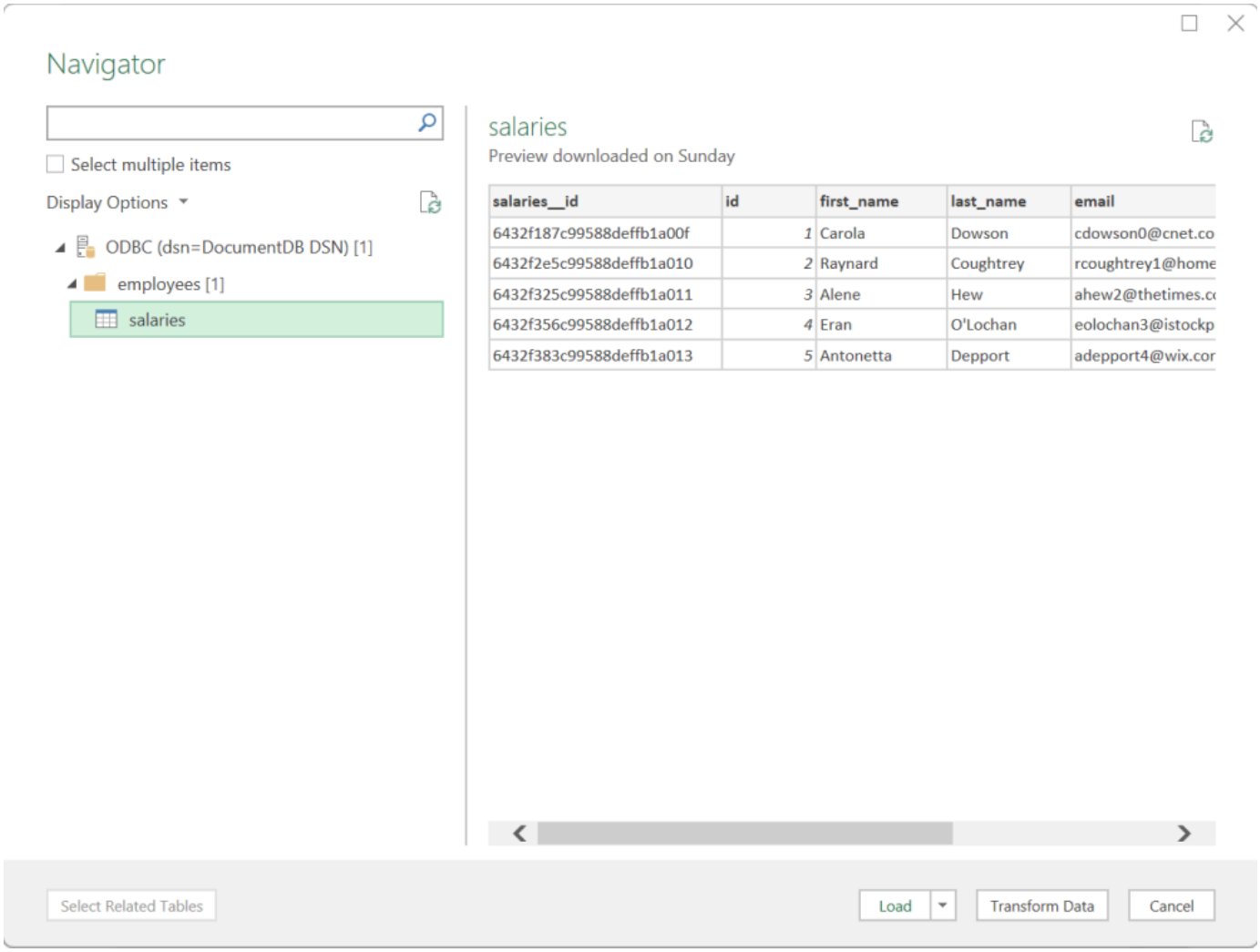
1. Stellen Sie sicher, dass der Amazon DocumentDB-Treiber korrekt installiert und konfiguriert wurde. Weitere Informationen finden Sie unter Einrichten des ODBC-Treibers in Windows.
2. Starten Sie Microsoft Excel.
3. Navigieren Sie zu Daten > Daten abrufen > Aus anderen Quellen.
4. Wählen Sie aus ODBC:



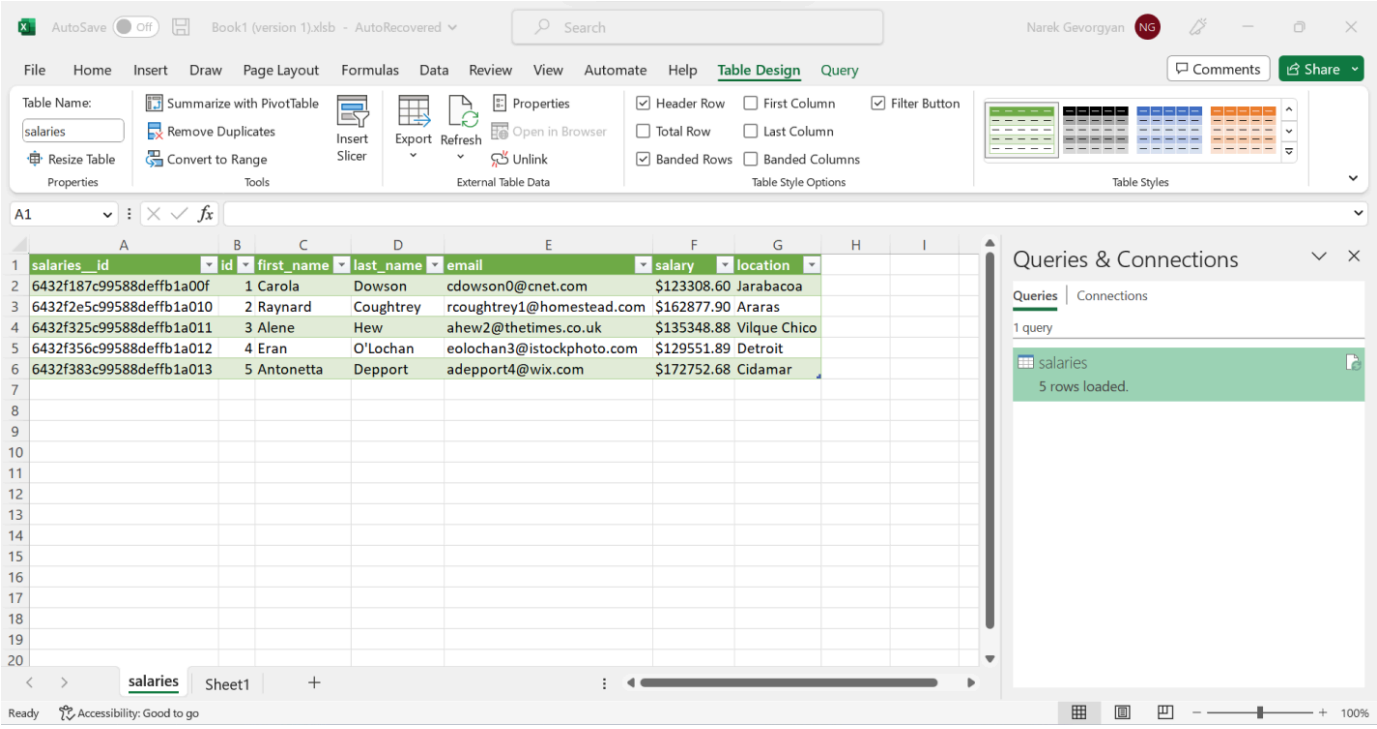
5. Wählen Sie im Drop-down-Menü Datenquellenname (DSN) die Datenquelle aus, die mit Amazon DocumentDB verknüpft ist:



6. Wählen Sie die Sammlung aus, aus der Sie Daten in Excel laden möchten:



7. Daten in Excel laden:



Stellen Sie von Microsoft Power BI Desktop aus eine Connect zu Amazon DocumentDB her

Themen

- Voraussetzungen
- Benutzerdefinierter Microsoft Power BI Desktop-Connector hinzufügen
- Verbindung mit dem benutzerdefinierten Amazon DocumentDB DocumentDB-Connector herstellen
- Konfiguration von Microsoft Power BI Gateway

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass der Amazon DocumentDB DocumentDB-ODBC-Treiber korrekt installiert ist.

Benutzerdefinierter Microsoft Power BI Desktop-Connector hinzufügen

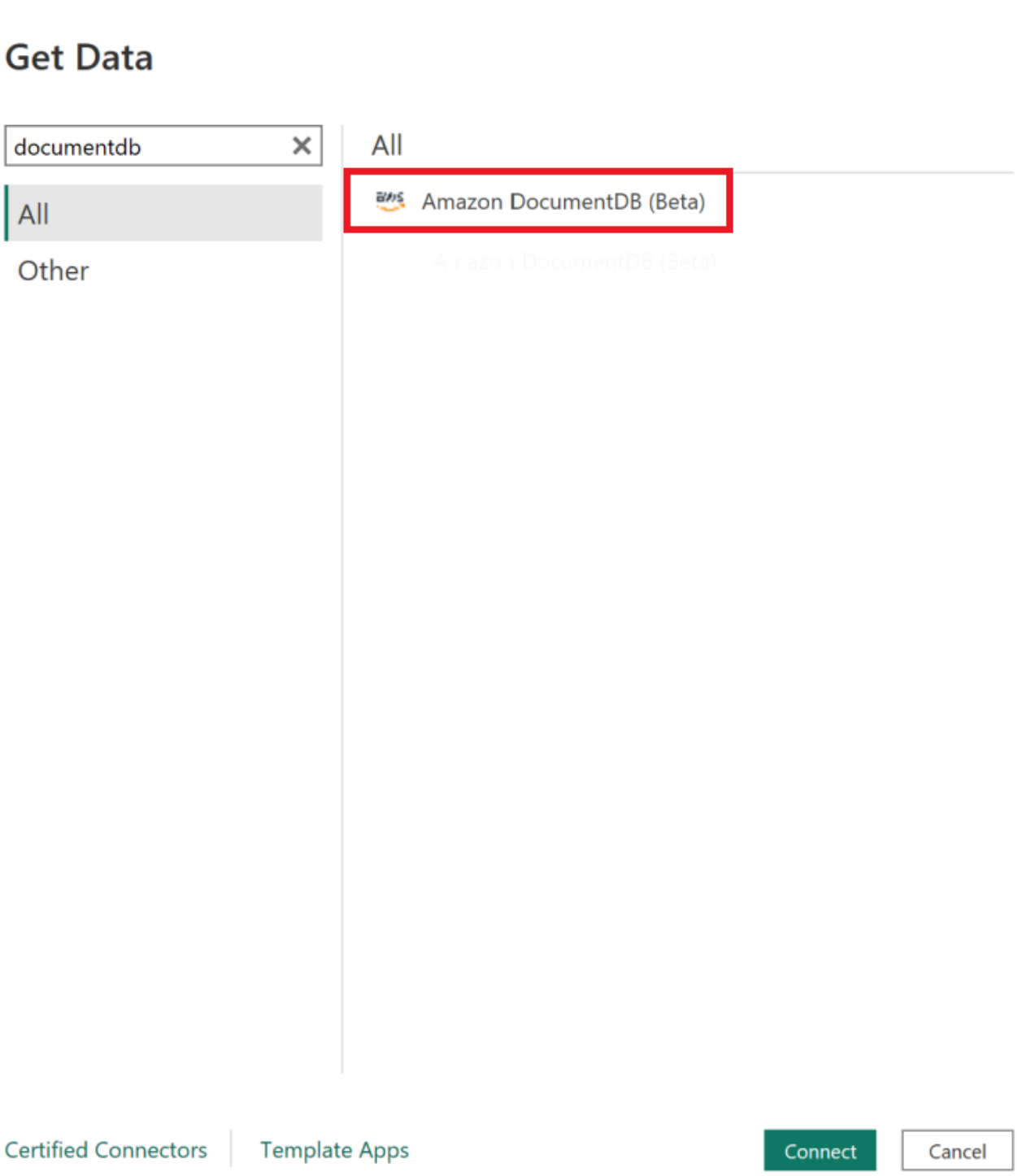
Kopieren Sie die`AmazonDocumentDBConnector.mez` Datei in den`<User>\Documents \Power BI Desktop\Custom Connectors\` Ordner (oder in den Ordner,`<User>\OneDrive \Documents\Power BI Desktop\Custom Connectors` falls Sie ihn verwenden OneDrive).

Dadurch kann Power BI auf den benutzerdefinierten Connector zugreifen. Den Connector zu Power BI Desktop erhalten Sie hier. Starten Sie Power BI Desktop neu, um sicherzustellen, dass der Connector geladen ist.

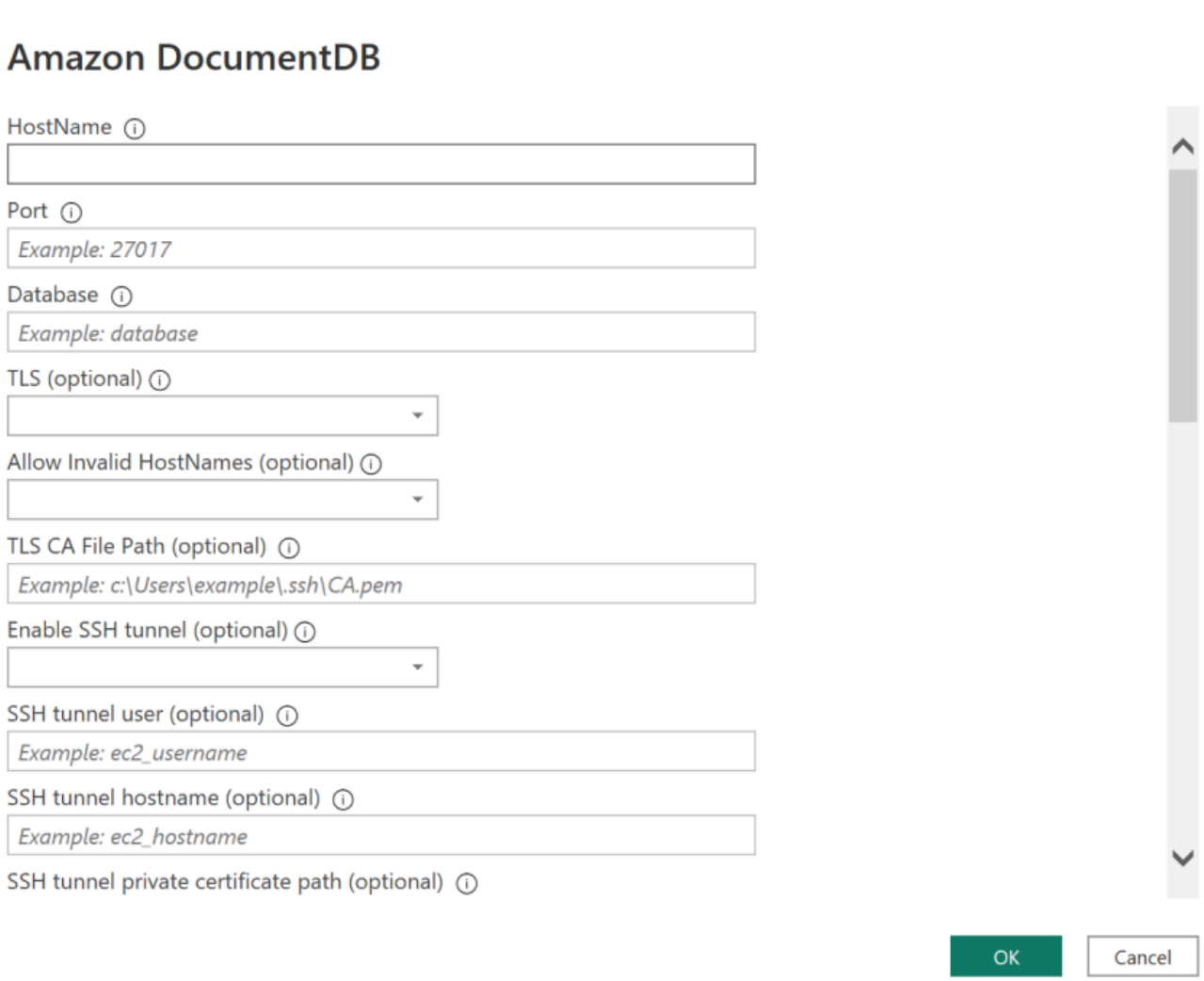
> **Note**
>
> Der benutzerdefinierte Connector unterstützt nur den Amazon DocumentDB DocumentDB-Benutzernamen und das Passwort für die Authentifizierung.

Verbindung mit dem benutzerdefinierten Amazon DocumentDB DocumentDB-Connector herstellen

1. Wählen Sie Amazon DocumentDB (Beta) unter Get Data aus und klicken Sie auf Connect. Wenn Sie wegen der Nutzung eines Drittanbieterdienstes eine Warnung erhalten, klicken Sie auf Weiter.



2. Geben Sie alle notwendigen Informationen ein, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, und klicken Sie dann auf OK:

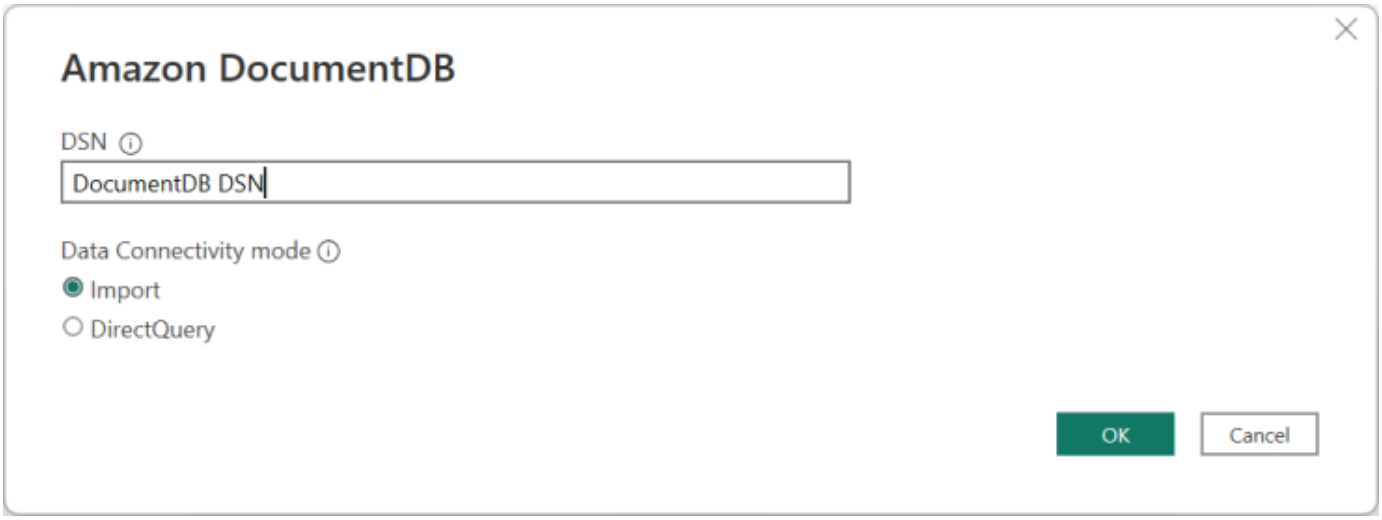


> **Note**
>
> Abhängig von der Konfiguration des Datenquellennamens (DSN) Ihres ODBC-Treibers wird der Bildschirm mit den SSH-Verbindungsdetails möglicherweise nicht angezeigt, wenn Sie die erforderlichen Informationen bereits in den DSN-Einstellungen angegeben haben.

3. Wählen Sie den Datenkonnektivitätsmodus:

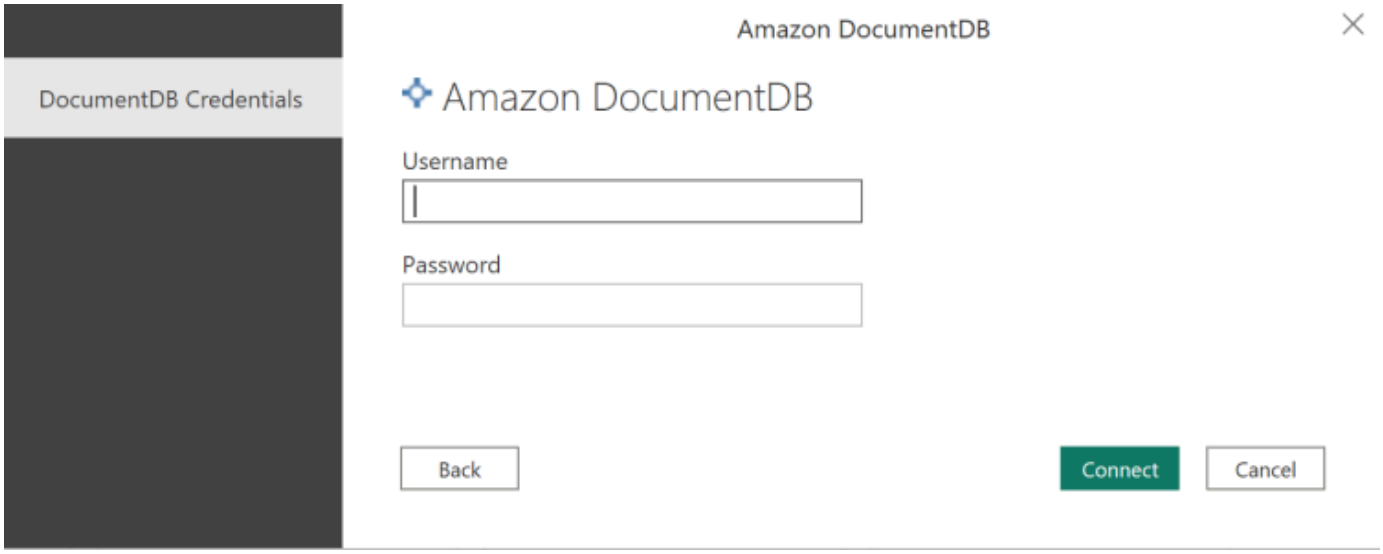
   - Import — lädt alle Daten und speichert die Informationen auf der Festplatte. Die Daten müssen aktualisiert und neu geladen werden, damit Datenaktualisierungen angezeigt werden.

- Direkte Abfrage — lädt keine Daten, sondern führt Live-Abfragen der Daten durch. Das bedeutet, dass Daten nicht aktualisiert und neu geladen werden müssen, um Datenaktualisierungen anzuzeigen.

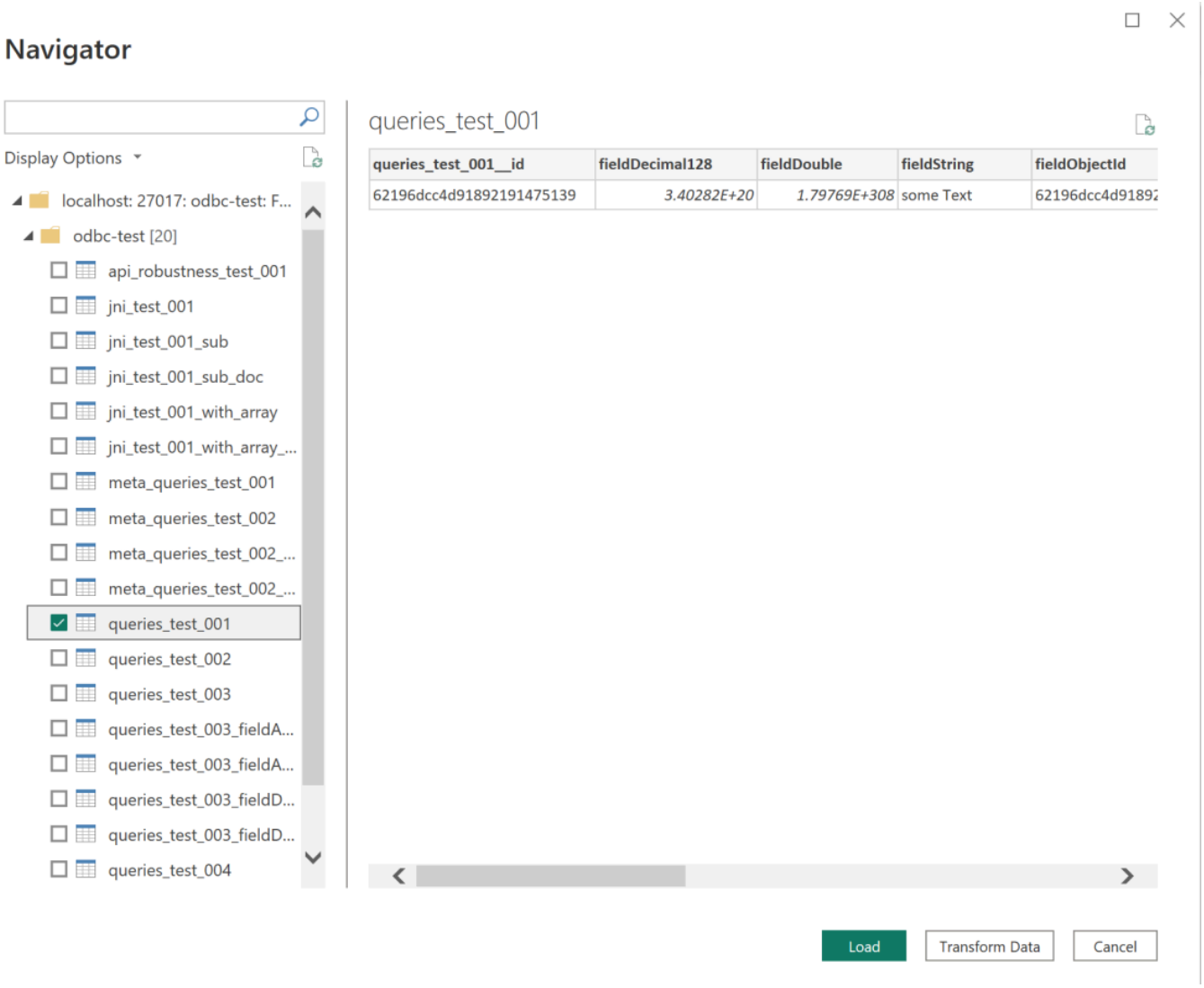


> **Note**
>
> Wenn Sie einen sehr großen Datensatz verwenden, kann das Importieren aller Daten länger dauern.

4. Wenn Sie zum ersten Mal eine Verbindung zu dieser Datenquelle herstellen, wählen Sie den Authentifizierungstyp aus und geben Sie Ihre Anmeldeinformationen ein, wenn Sie dazu aufgefordert werden. Klicken Sie dann auf Connect:



5. Wählen Sie im Navigator-Dialogfeld die gewünschten Datenbanktabellen aus und klicken Sie dann entweder auf Laden, um die Daten zu laden, oder auf Daten transformieren, um mit der Transformation der Daten fortzufahren.



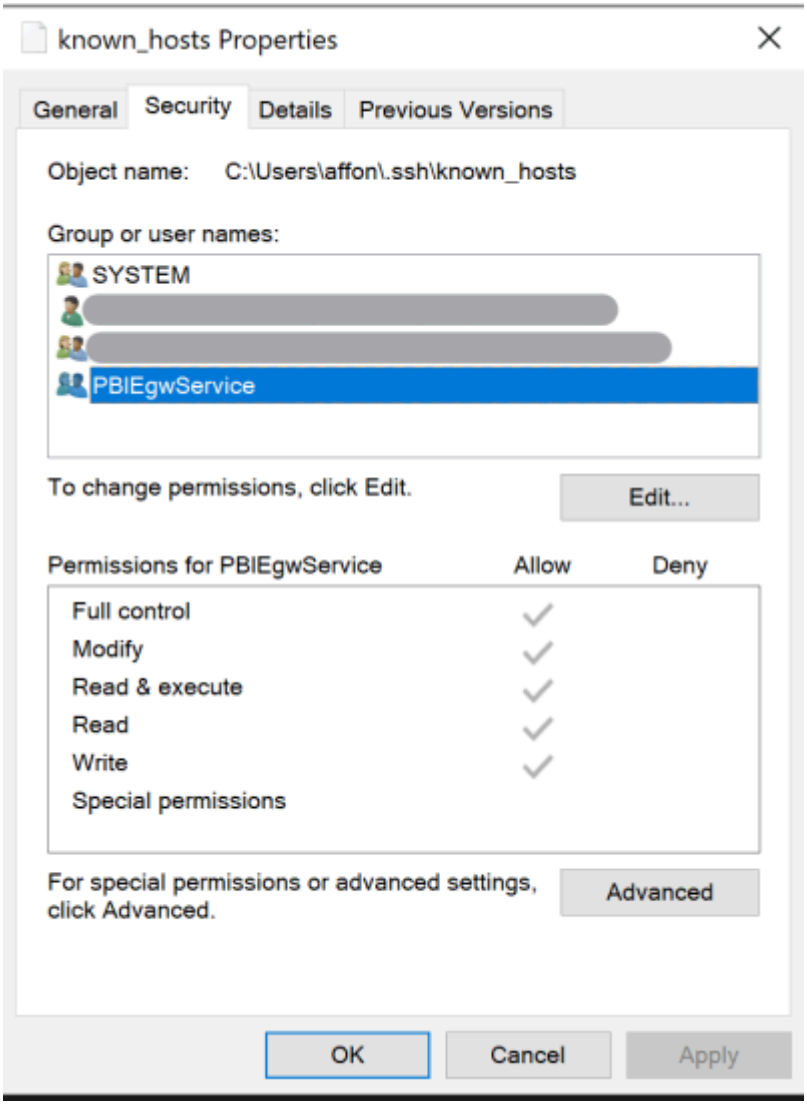
> **Note**
>
> Ihre Datenquelleneinstellungen werden gespeichert, sobald Sie eine Verbindung herstellen. Um sie zu ändern, wählen Sie Daten transformieren > Datenquelleneinstellungen.

Konfiguration von Microsoft Power BI Gateway

Voraussetzungen:

- Stellen Sie sicher, dass der benutzerdefinierte Connector mit Power BI Gateway funktioniert.
- Stellen Sie sicher, dass der ODBC-DSN in den ODBC-Datenquellen auf der Registerkarte System auf dem Computer erstellt wurde, auf dem Power BI Gateway installiert ist.

Wenn Sie die interne SSH-Tunnelfunktion verwenden,`known_hosts` muss sich die Datei dort befinden, wo das Power BI-Dienstkonto Zugriff darauf hat.



> **Note**
>
> Dies gilt auch für alle Dateien, die Sie möglicherweise benötigen, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, z. B. eine Certificate Authority (CA) - Zertifikatsdatei (PEM-Datei).

Automatische Schemagenerierung

Der ODBC-Treiber verwendet den Amazon DocumentDB DocumentDB-JDBC-Treiber über JNI (Java Native Interface), sodass die automatische Schemagenerierung im JDBC-Treiber ähnlich funktioniert. Weitere Informationen zur automatischen Schemagenerierung finden Sie unter Automatische JDBC-Schemagenerierung. Um mehr über die ODBC-Treiberarchitektur zu erfahren, klicken Sie außerdem hier.

SQL-Unterstützung und Einschränkungen

Der Amazon DocumentDB DocumentDB-ODBC-Treiber ist ein schreibgeschützter Treiber, der eine Teilmenge von SQL-92 und einige gängige Erweiterungen unterstützt. Weitere Informationen finden Sie in der Dokumentation zur ODBC-Unterstützung und zu den Einschränkungen.

Fehlerbehebung

Wenn Sie Probleme bei der Verwendung des Amazon DocumentDB-ODBC-Treibers haben, lesen Sie den Leitfaden zur Fehlerbehebung.

Kontingente und Limits für Amazon DocumentDB

In diesem Thema werden die Ressourcenkontingente, Limits und Benennungseinschränkungen für Amazon DocumentDB (mit MongoDB-Kompatibilität) beschrieben.

Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) und Amazon Neptune gemeinsam genutzt wird.

Themen

Unterstützte -Instance-Typen

Amazon DocumentDB unterstützt On-Demand-Instances und die folgenden Instance-Typen:

- Speicheroptimiert
  - R6G-Instance-Typen: `db.r6g.large`, `db.r6g.2xlarge`, `db.r6g.4xlarge`, `db.r6g.8xlarge`, `db.r6g.12xlarge`, `db.r6g.16xlarge`.
  - R5-Instance-Typen: `db.r5.large`, `db.r5.2xlarge`, `db.r5.4xlarge`, `db.r5.8xlarge`, `db.r5.12xlarge`. `db.r5.16xlarge` `db.r5.24xlarge`
  - R4-Instance-Typen: `db.r4.large`, `db.r4.2xlarge`, `db.r4.4xlarge`, `db.r4.8xlarge`, `db.r4.16xlarge`.
- Spitzenlastleistung:

- T4G-Instance-Typen: `db.t4g.medium`.
- T3-Instance-Typen: `db.t3.medium`.

Weitere Informationen zu den unterstützten Instance-Typen und deren Spezifikationen finden Sie unter Instance-Klassen-Spezifikationen.

Unterstützte Regionen

Amazon DocumentDB ist in den folgenden AWS Regionen verfügbar:

Name der Region	Region	Availability Zones (Datenverarbeitung)
USA Ost (Ohio)	`us-east-2`	3
USA Ost (Nord-Virginia)	`us-east-1`	6
USA West (Oregon)	`us-west-2`	4
Südamerika (São Paulo)	`sa-east-1`	3
Asien-Pazifik (Hongkong)	`ap-east-1`	3
Asien-Pazifik (Hyderabad)	`ap-south-2`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	4
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Kanada (Zentral)	`ca-central-1`	3
Region China (Peking)	`cn-north-1`	3

Name der Region	Region	Availability Zones (Datenverarbeitung)
China (Ningxia)	`cn-northwest-1`	3
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
AWS GovCloud (USA West)	`us-gov-west-1`	3
AWS GovCloud (USA-Ost)	`us-gov-east-1`	3

Regionale Kontingente

Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird. Die folgende Tabelle enthält regionale Limits, die von Amazon DocumentDB und Amazon RDS gemeinsam genutzt werden.

Die folgenden Limits gelten pro AWS Konto und Region.

Ressource	AWS Standardlimit
Cluster	40
Cluster-Parametergruppen	50
Ereignisabonnements	20
Instances	40

Ressource	AWS Standardlimit
Manuelle Cluster-Snapshots	100
Read Replicas pro Cluster	15
Subnetzgruppen	50
Subnetze pro Subnetzgruppe	20
Tags pro Ressource	50
VPC-Sicherheitsgruppen pro Instance	5

Sie können mit Service Quotas eine Erhöhung für ein Kontingent beantragen, sofern das Kontingent anpassbar ist. Einige Anfragen werden automatisch aufgelöst, andere werden an übermittelt AWS Support. Sie können den Status einer Kontingenterhöhung verfolgen, die an übermittelt wird AWS Support. Anfragen zur Erhöhung der Servicekontingente erhalten keinen bevorzugten Support. Wenn Sie eine dringende Anfrage haben, wenden Sie sich bitte an AWS Support. Weitere Informationen zu Service Quotas finden Sie unter What Is Service Quotas?.

So fordern Sie eine Kontingenterhöhung für Amazon DocumentDB an:

1. Öffnen Sie die Service Quotas-Konsole unter https://console.aws.amazon.com/servicequotas und melden Sie sich bei Bedarf an.
2. Wählen Sie im Navigationsbereich AWS -Services.
3. Wählen Sie Amazon DocumentDB aus der Liste aus oder geben Sie Amazon DocumentDB in das Suchfeld ein.
4. Wenn das Kontingent einstellbar ist, können Sie sein Optionsfeld oder seinen Namen auswählen und dann rechts oben auf der Seite die Option Request quota increase (Kontingenterhöhung beantragen) auswählen.
5. Geben Sie unter Change quota value (Kontingentwert ändern) den neuen Wert ein. Der neue Wert muss größer als der aktuelle Wert sein.
6. Wählen Sie Request (Anfrage). Nachdem die Anfrage genehmigt wurde, wird Applied quota value (Angewandter Kontingentwert) für das Kontingent auf den neuen Wert eingestellt.
7. Um ausstehende oder kürzlich genehmigte Anfragen anzuzeigen, wählen Sie im Navigationsbereich die Option Dashboard . Wählen Sie für ausstehende Anfragen den Status

der Anfrage, um die Anfrage zu öffnen. Der Anfangsstatus einer Anfrage ist `Pending`. Nachdem sich der Status in geändert hat`Quota requested`, sehen Sie die Fallnummer mit AWS Support. Wählen Sie die Fallnummer, um das Ticket für Ihre Anfrage zu öffnen.

Aggregationsbeschränkungen

In der folgenden Tabelle werden Aggregationslimits in Amazon DocumentDB beschrieben.

Ressource	Limit
Maximale Anzahl der unterstützten Stages	500

Cluster-Beschränkungen

In der folgenden Tabelle werden die Instance-basierten Cluster-Limits von Amazon DocumentDB beschrieben.

Ressource	Limit
Cluster-Größe (Summe aller Sammlungen und Indizes)	128 TiB
Sammlungsgröße (Summe aller Sammlungen darf die Cluster-Beschränkung nicht überschreiten) – beinhaltet nicht die Indexgröße.	32 TB
Sammlungen pro Cluster	100 000
Datenbanken pro Cluster	100 000
Datenbankgröße (Summe aller Datenbanken darf die Cluster-B eschränkung nicht überschreiten)	128 TiB
Verschachtelungstiefe für Dokumente	200 Ebenen
Dokumentengröße	16 MB
Indexschlüsselgröße	2048 Bytes
Indizes pro Sammlung	64
Schlüssel in einem zusammengesetzten Index	32

Ressource	Limit
Maximale Anzahl der Schreibvorgänge in einem einzelnen Stapelbefehl	100 000
Anzahl Benutzer pro Cluster	1000

Instance-Limits

In der folgenden Tabelle werden Amazon DocumentDB-Limits pro Instance beschrieben.

Instance-Typ	Instance-Speicher (GiB)	Verbindungen (alle)	Cursor-Limit	Offene Transaktionen	Verbindungen (aktiv)
T3.medium	4	500	30	50	102
T4G .medium	4	500	30	50	102
R4.large	15,25	1700	450	N/A	1100
R4.xlarge	30,5	3400	450	N/A	2700
R4.2xlarge	61	6800	450	N/A	4500
R4.4xlarge	122	13600	725	N/A	4500
R4.8xlarge	288	27200	1450	N/A	4500
R4.16xlarge	488	30000	2900	N/A	4500
R5.large	16	1700	450	200	1100
R5.xlarge	32	3500	450	400	2700
R5.2xlarge	64	7100	450	800	4500
R5.4xlarge	128	14200	760	1600	4500
R5.8xlarge	256	28400	1520	3200	4500
R5.12xlarge	383	30000	2280	4800	4500
R5.16xlarge	512	30000	3040	6400	4500
R5.24xlarge	768	30000	4560	9600	4500
R6G .large	16	1700	450	200	1100

Instance-Typ	Instance-Speicher (GiB)	Verbindungen (alle)	Cursor-Limit	Offene Transaktionen	Verbindungen (aktiv)
R6G .xlarge	32	3500	450	400	2700
R6G .2xlarge	64	7100	450	800	4500
R6G .4xlarge	128	14200	760	1600	4500
R6G .8xlarge	256	28400	1520	3200	4500
R6G .12xlarge	383	30000	2280	4800	4500
R6G .16xlarge	512	30000	3040	6400	4500

Mit den folgenden CloudWatch Metriken können Sie die Limits pro Instance überwachen und einen Alarm auslösen. Weitere Informationen zu Amazon DocumentDB CloudWatch -Metriken finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.

Limit	CloudWatch Metriken
Instance-Speicher	FreeableMemory
Verbindungen	DatabaseConnectionsMax
Cursor	DatabaseCursorsMax
Transaktionen	TransactionsOpenMax

Benennungseinschränkungen

In der folgenden Tabelle werden Benennungseinschränkungen in Amazon DocumentDB beschrieben.

Ressource	Standardlimit
Cluster Identifier (Cluster-Kennung)	• Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten. • Muss für alle Cluster (über Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.
Name der Sammlung: `<col>`	Die Länge beträgt [1–57] Zeichen.
Datenbankname: `<db>`	Die Länge beträgt [1–63] Zeichen.
Vollqualifizierter Sammlungsname: `<db>.<col>`	Die Länge beträgt [3–120] Zeichen.
Vollqualifizierter Indexname: `<db>.<col>.$<index>`	Die Länge beträgt [6–127] Zeichen.
Indexname: `<col>$<index>`	Die Länge beträgt [3–63] Zeichen.
Instance-ID	• Die Länge beträgt [1–63] Buchstaben, Zahlen oder Bindestriche

Ressource	Standardlimit
	• Muss mit einem Buchstaben beginnen • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten • Muss für alle Instances (über Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

Hauptpasswort	• Länge beträgt [8–100] druckbare ASCII-Zeichen.
	• Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
	  • **/** (Schrägstrich)
	  • **"** (doppeltes Anführungszeichen)
	  • **@** ('At'-Symbol)

Masterbenutzername	• Die Länge beträgt [1 bis 63] alphanumerische Zeichen. • Muss mit einem Buchstaben beginnen. • Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

Ressource	Standardlimit
Parametergruppenname	• Die Länge muss [1 bis 255] alphanumerische Zeichen betragen. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

TTL-Einschränkungen

Es kann nicht garantiert werden, dass Löschoperationen über einem TTL-Index innerhalb eines bestimmten Zeitraums abgeschlossen und mit höchster Priorität behandelt werden. Faktoren wie Ressourcenauslastung der Instance, Dokumentgröße und Gesamtdurchsatz können Einfluss auf die Dauer von TTL-Löschoperationen haben.

Elastic-Cluster-Limits

Die folgende Tabelle beschreibt die maximalen Limits in elastischen Amazon DocumentDB-Clustern.

Ressource	Limit
Elastic Cluster pro Region	20
vCPU, summiert über alle elastischen Cluster pro Region	1024
Manuelle Cluster-Snapshots pro Region	20
Shards pro Cluster	32
Speicher pro Cluster (wenn Daten gleichmäßig nach Shard-Schlüssel verteilt werden)	4 PiB

Ressource	Limit
Verbindungen zum Cluster	Der niedrigere Wert von 300.000 oder die Anzahl der Shards x Verbindungslimit, das der vCPU pro Shard zugeordnet ist
UnSharded Sammlungsgröße	32TB
Größe der Sammlung mit Sharding (wenn Daten gleichmäßig nach Shard-Schlüssel verteilt werden)	1PB
Datenbanken pro Cluster	10.000
UnSharded Sammlungen pro Cluster	100 000
Sharded-Sammlungen pro Cluster	1000
Benutzer pro Cluster	100
Schreibvorgänge in einem einzigen Batch-Befehl	100 000
Indizes pro Sammlung	64
Verschachtelungstiefe für Dokumente	100 Ebenen
Dokumentengröße	16 MB
Indexschlüsselgröße	2048 Bytes
Schlüssel in einem zusammengesetzten Index	32

Elastic-Cluster-Shard-Limits

Die folgende Tabelle beschreibt die maximalen Shard-Limits in elastischen Amazon DocumentDB-Clustern.

Ressource	Limit
vCPU pro Shard-Instance	64
Instances pro Shard	16
Zeichenfolge pro Shard	128 TiB
Speicher pro Sammlung und Shard	32TB

Elastic-Cluster-CPU-, Arbeitsspeicher-, Verbindungs- und Cursor-Limits pro Shard

Die folgende Tabelle beschreibt die maximalen CPU-, Arbeitsspeicher-, Verbindungs- und Cursor-Limits in elastischen Amazon DocumentDB-Cluster-Shards.

vCPUs pro Shard	Instance-Speicher (GiB)	Verbindungslimit	Cursor-Limit
2	16	1700	450
4	32	3500	450
8	64	7100	450
16	128	14200	760
32	256	28400	1520
48	383	30000	2280
64	512	30000	3040

Abfragen

In diesem Abschnitt werden alle Aspekte der Abfrage mit Amazon DocumentDB erläutert.

Themen

Abfragen von Dokumenten

Manchmal müssen Sie möglicherweise den Bestand Ihres Online-Shops nachschlagen, damit Kunden das Angebot sehen und kaufen können. Die Abfrage einer Sammlung ist relativ einfach, unabhängig davon, ob Sie alle Dokumente in der Sammlung haben möchten oder nur die Dokumente, die ein bestimmtes Kriterium erfüllen.

Verwenden Sie die Operation `find()`, um Dokumente abzufragen. Der Befehl `find()` hat einen einzigen Dokumentenparameter, der die Kriterien für die Auswahl der zurückzugebenden Dokumente definiert. Die Ausgabe von `find()` ist ein Dokument, das als einzelne Textzeile ohne Zeilenumbrüche formatiert ist. Um das Ausgabedokument für eine bessere Lesbarkeit zu formatieren, verwenden Sie `find().pretty()`. Alle Beispiele in diesem Thema verwenden `.pretty()` zum Formatieren der Ausgabe.

Die folgenden Codebeispiele verwenden die vier Dokumente, die Sie in den beiden vorherigen Übungen in die `example` Sammlung eingefügt haben – `insertOne()` und `insertMany()` die sich im Abschnitt Hinzufügen von Dokumenten unter Arbeiten mit Dokumenten befinden.

Themen

- Abrufen von Dokumenten, die einem Array entsprechen
- Abrufen von Dokumenten, die einem Wert in einem Array entsprechen
- Abrufen von Dokumenten mithilfe von Operatoren

Abrufen aller Dokumente in einer Sammlung

Um alle Dokumente in Ihrer Sammlung abzurufen, verwenden Sie die Operation `find()` mit einem leeren Abfragedokument.

Die folgende Abfrage gibt alle Dokumente der Sammlung `example` zurück.

```
db.example.find( {} ).pretty()
```

Abrufen von Dokumenten, die einem Feldwert entsprechen

Um alle Dokumente abzurufen, die mit einem Feld und einem Wert übereinstimmen, verwenden Sie die Operation `find()` mit einem Abfragedokument, das die entsprechenden Felder und Werte identifiziert.

Bei Verwendung der vorangegangenen Dokumente gibt diese Abfrage alle Dokumente zurück, bei denen das Feld "Item" (Element) "Pen" (Stift) entspricht.

```
db.example.find( { "Item": "Pen" } ).pretty()
```

Abrufen von Dokumenten, die mit einem eingebetteten Dokument übereinstimmen

Um alle Dokumente zu suchen, die mit einem eingebetteten Dokument übereinstimmen, verwenden Sie die Operation `find()` mit einem Abfragedokument, in dem der Name des eingebetteten Dokuments sowie alle Felder und Werte für dieses eingebettete Dokument angegeben werden.

Beim Vergleichen mit einem eingebetteten Dokument muss das eingebettete Dokument denselben Namen haben wie in der Abfrage. Zudem müssen die Felder und Werte im eingebetteten Dokument mit der Abfrage übereinstimmen.

Die folgende Abfrage gibt nur das Dokument "Poster Paint" zurück. Dies liegt daran, dass "Pen" über verschiedene Werte für "`OnHand`" und "`MinOnHand`" verfügt und "Spray Paint" ein weiteres Feld (`OrderQnty`) als das Abfragedokument besitzt.

```
db.example.find({"Inventory": {
    "OnHand": 47,
    "MinOnHand": 50 } } ).pretty()
```

Abrufen von Dokumenten, die einem Feldwert in einem eingebetteten Dokument entsprechen

Um alle Dokumente zu suchen, die mit einem eingebetteten Dokument übereinstimmen, verwenden Sie die Operation `find()` mit einem Abfragedokument, in dem der Name des eingebetteten Dokuments sowie alle Felder und Werte für dieses eingebettete Dokument angegeben werden.

Aufgrund der vorangegangenen Dokumente verwendet die folgende Abfrage "Punktnotation", um das eingebettete Dokument und die Felder von Interesse anzugeben. Jedes Dokument, das damit übereinstimmt, wird zurückgegeben, unabhängig davon, welche anderen Felder im eingebetteten Dokument vorhanden sind. Die Abfrage gibt "Poster Paint" und "Spray Paint" zurück, weil sie beide den angegebenen Feldern und Werten entsprechen.

```
db.example.find({"Inventory.OnHand": 47, "Inventory.MinOnHand": 50 }).pretty()
```

Abrufen von Dokumenten, die einem Array entsprechen

Um alle Dokumente zu finden, die einem Array entsprechen, verwenden Sie die Operation `find()` mit dem Namen des Arrays, an dem Sie interessiert sind, und allen Werten in diesem Array. Die Abfrage gibt alle Dokumente zurück, in denen sich ein Array mit diesem Namen befindet und in denen die Array-Werte identisch sind und die gleiche Reihenfolge wie in der Abfrage aufweisen.

Die folgende Abfrage gibt nur "Pen" zurück, da "Poster Paint" über eine zusätzlichen Farbe (White) verfügt und die Farben in "Spray Paint" in einer anderen Reihenfolge vorliegen.

```
db.example.find( { "Colors": ["Red","Green","Blue","Black"] } ).pretty()
```

Abrufen von Dokumenten, die einem Wert in einem Array entsprechen

Um alle Dokumente mit einem bestimmten Array-Wert zu finden, verwenden Sie die Operation `find()` mit dem Namen und Wert des Arrays, an dem Sie interessiert sind.

```
db.example.find( { "Colors": "Red" } ).pretty()
```

Bei der vorherigen Operation werden alle drei Dokumente zurückgegeben, da jedes davon ein Array mit dem Namen `Colors` und den Wert "`Red`" irgendwo im Array besitzt. Wenn Sie den Wert "`White`" angeben, gibt die Abfrage nur "Poster Paint" zurück.

Abrufen von Dokumenten mithilfe von Operatoren

Die folgende Abfrage gibt alle Dokumente zurück, in denen der Wert "`Inventory.OnHand`" kleiner als 50 ist.

```
db.example.find(
        { "Inventory.OnHand": { $lt: 50 } } )
```

Eine Liste der unterstützten Abfrageoperatoren finden Sie unter Abfrage- und Projektions-Operatoren.

Abfrageplan

Wie kann ich die **`executionStats`** für einen Abfrageplan anzeigen?

Wenn Sie ermitteln, warum eine Abfrage langsamer als erwartet ausgeführt wird, kann es hilfreich sein zu verstehen, was die `executionStats` für den Abfrageplan sind. Die `executionStats` geben die Anzahl der Dokumente an, die von einer bestimmten Stufe zurückgegeben wurden (`nReturned`), die in jeder Stufe verbrachte Ausführungszeit (`executionTimeMillisEstimate`) und die Zeit, die zum Generieren eines Abfrageplans benötigt wird (`planningTimeMillis`). Sie können die zeitintensivsten Stufen Ihrer Abfrage bestimmen, um Ihre Optimierungsbemühungen aus der Ausgabe von `executionStats` zu konzentrieren, wie in den Abfragebeispielen unten gezeigt. Der `executionStats`-Parameter unterstützt derzeit keine `update`- und `delete`-Befehle.

> **Note**
>
> Amazon DocumentDB emuliert die MongoDB 3.6-API auf einer speziell entwickelten Datenbank-Engine, die ein verteiltes, fehlertolerantes, selbstverstärkendes Speichersystem verwendet. Daher `explain()` können sich Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterscheiden. Kunden, die die Kontrolle über ihren Abfrageplan wünschen, können den `$hint`-Operator verwenden, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Führen Sie die Abfrage, die Sie verbessern möchten, unter dem Befehl `explain()` wie folgt aus.

```
db.runCommand({explain: {query document}}).
explain("executionStats").executionStats;
```

Im Folgenden finden Sie eine Beispieloperation.

```
db.fish.find({}).limit(2).explain("executionStats");
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "test.fish",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "LIMIT_SKIP",
                "inputStage" : {
                    "stage" : "COLLSCAN"
                }
            }
        }
    },
    "executionStats" : {
        "executionSuccess" : true,
        "executionTimeMillis" : "0.063",
        "planningTimeMillis" : "0.040",
        "executionStages" : {
            "stage" : "SUBSCAN",
            "nReturned" : "2",
            "executionTimeMillisEstimate" : "0.012",
            "inputStage" : {
                "stage" : "LIMIT_SKIP",
                "nReturned" : "2",
                "executionTimeMillisEstimate" : "0.005",
                "inputStage" : {
                    "stage" : "COLLSCAN",
                    "nReturned" : "2",
                    "executionTimeMillisEstimate" : "0.005"
                }
            }
        }
```

```
    },
    "serverInfo" : {
        "host" : "enginedemo",
        "port" : 27017,
        "version" : "3.6.0"
    },
    "ok" : 1
}
```

Wenn Sie nur die `executionStats` aus der obigen Abfrage sehen möchten, können Sie den folgenden Befehl verwenden. Bei kleinen Sammlungen kann der Amazon DocumentDB-Abfrageprozessor wählen, keinen Index zu verwenden, wenn die Leistungssteigerungen vernachlässigbar sind.

```
db.fish.find({}).limit(2).explain("executionStats").executionStats;
```

Abfrageplan-Cache

Um die Leistung zu optimieren und die Planungsdauer zu reduzieren, speichert Amazon DocumentDB Abfragepläne intern zwischen. Auf diese Weise können Abfragen mit derselben Form direkt mit einem zwischengespeicherten Plan ausgeführt werden.

Dieses Caching kann jedoch manchmal zu einer zufälligen Verzögerung für dieselbe Abfrage führen. Beispielsweise kann eine Abfrage, die normalerweise eine Sekunde dauert, gelegentlich zehn Sekunden dauern. Dies liegt daran, dass die Reader-Instance im Laufe der Zeit verschiedene Formen der Abfrage zwischengespeichert hat und somit Speicher verbraucht. Wenn Sie diese zufällige Langsamkeit feststellen, müssen Sie nichts unternehmen, um den Speicher freizugeben. Das System verwaltet die Speichernutzung für Sie. Sobald der Speicher einen bestimmten Schwellenwert erreicht hat, wird er automatisch freigegeben.

Erläutern der Ergebnisse

Wenn Sie Informationen zu Abfrageplänen zurückgeben möchten, unterstützt Amazon DocumentDB den Ausführlichkeitsmodus `queryPlanner`. Die `explain` Ergebnisse geben den vom Optimierer ausgewählten Abfrageplan in einem Format zurück, das dem folgenden ähnelt:

```
{
   "queryPlanner" : {
      "plannerVersion" : <int>,
```

```
      "namespace" : <string>,
      "winningPlan" : {
         "stage" : <STAGE1>,
         ...
         "inputStage" : {
            "stage" : <STAGE2>,
            ...
            "inputStage" : {
               ...
            }
         }
      }
   }
}
```

In den folgenden Abschnitten werden allgemeine `explain` Ergebnisse definiert.

Themen

- Scan- und Filterphase
- Indexüberschneidung
- Indexunion
- Mehrere Index-Überschneidung/-Verknüpfung
- Zusammengesetzter Index
- Sortierphase
- Gruppenphase

Scan- und Filterphase

Der Optimierer kann einen der folgenden Scans auswählen:

COLLSCAN

Diese Phase ist ein sequenzieller Sammlungsscan.

```
{
    "stage" : "COLLSCAN"
}
```

IXSCAN

Diese Phase scannt die Indexschlüssel. Der Optimierer kann das Dokument innerhalb dieser Phase abrufen, was dazu führen kann, dass später eine FETCH-Phase angehängt wird.

```
db.foo.find({"a": 1})
{
    "stage" : "IXSCAN",
    "direction" : "forward",
    "indexName" : <idx_name>
}
```

FETCH

Wenn der Optimierer Dokumente in einer anderen Phase als IXSCAN abgerufen hat, enthält das Ergebnis eine FETCH-Phase. Die obige IXSCAN-Abfrage kann beispielsweise zu einer Kombination von FETCH- und IXSCAN-Phasen führen:

```
db.foo.find({"a": 1})
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXSCAN",
        "indexName" : <idx_name>
    }
}
```

IXONLYSCAN scannt nur den Indexschlüssel. Das Erstellen zusammengesetzter Indizes vermeidet FETCH nicht.

Indexüberschneidung

IXAND

Amazon DocumentDB kann eine IXAND-Stufe mit einem inputStages-Array von IXSCAN enthalten, wenn es eine Indexüberschneidung verwenden kann. Beispielsweise wird möglicherweise eine Ausgabe wie folgt angezeigt:

```
{
    "stage" : "FETCH",
```

```
    "inputStage" : {
        "stage" : "IXAND",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Indexunion

IXOR

Ähnlich wie bei einer Indexüberschneidung kann Amazon DocumentDB eine `IXOR` Stufe mit einem `inputStages`Array für den `$or` Operator enthalten.

```
db.foo.find({"$or": [{"a": {"$gt": 2}}, {"b": {"$lt": 2}}]})
```

Für die obige Abfrage kann die Explain-Ausgabe wie folgt aussehen:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Mehrere Index-Überschneidung/-Verknüpfung

Amazon DocumentDB kann mehrere Indexüberschneidungs- oder Vereinigungsphasen miteinander kombinieren und dann das Ergebnis abrufen. Beispielsweise:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                ...
            },
            {
                "stage" : "IXAND",
                "inputStages" : [
                    {
                        "stage" : "IXSCAN",
                        ...
                    },
                    {
                        "stage" : "IXSCAN",
                        ...
                    }
                ]
            }
        ]
    }
}
```

Die Verwendung von Indexüberschneidungs- oder Vereinigungsphasen wird vom Indextyp (spärlich, zusammengesetzt usw.) nicht beeinflusst.

Zusammengesetzter Index

Die zusammengesetzte Indexnutzung von Amazon DocumentDB ist in den ersten Teilmengen indizierter Felder nicht begrenzt. Sie kann Index mit dem Suffixteil verwenden, ist aber möglicherweise nicht sehr effizient.

Beispielsweise `{ a: 1, b: -1 }` kann der zusammengesetzte Index von alle drei folgenden Abfragen unterstützen:

```
db.orders.find( { a: 1 } } )

db.orders.find( { b: 1 } } )

db.orders.find( { a: 1, b: 1 } } )
```

Sortierphase

Wenn es einen Index für den/die angeforderten Sortierschlüssel(e) gibt, kann Amazon DocumentDB den Index verwenden, um die Reihenfolge abzurufen. In diesem Fall enthält das Ergebnis keine -`SORT`Phase, sondern eine -`IXSCAN`Phase. Wenn der Optimierer eine einfache Sortierung bevorzugt, enthält er eine Stufe wie die folgende:

```
{
    "stage" : "SORT",
    "sortPattern" : {
        "a" : 1,
        "b" : -1
    }
}
```

Gruppenphase

Amazon DocumentDB unterstützt zwei verschiedene Gruppenstrategien:

- `SORT_AGGREGATE`: Auf dem Datenträgersortieraggregat.
- `HASH_AGGREGATE`: Im Speicher-Hash-Aggregat.

Abfragen von Geodaten mit Amazon DocumentDB

In diesem Abschnitt wird beschrieben, wie Sie Geodaten mit Amazon DocumentDB abfragen können. Nachdem Sie diesen Abschnitt gelesen haben, können Sie antworten, wie Geodaten in Amazon DocumentDB speichert, abfragt und indiziert.

Themen

Übersicht

Häufige Anwendungsfälle für Geospatial umfassen Näherungsanalysen aus Ihren Daten. Zum Beispiel „Alle Flughafen innerhalb von 50 Kilometern von Phoenix finden“ oder „Von einem bestimmten Ort aus den nächstgelegenen Ort finden“. Amazon DocumentDB verwendet die GeoJSON-Spezifikation, um Geodaten darzustellen. GeoJSON ist eine Open-Source-Spezifikation für die JSON-Formatierung von Formen in einem Koordinatenraum. GeoJSON-Koordinaten erfassen sowohl Längen- als auch Breitengrad und stellen Positionen auf einer erdähnlichen Kugel dar.

Indizieren und Speichern von Geodaten

Amazon DocumentDB verwendet den GeoJSON-Typ „Point“, um Geodaten zu speichern. Jedes GeoJSON-Dokument (oder Unterdokument) besteht im Allgemeinen aus zwei Feldern:

- Typ – die Form, die dargestellt wird, die Amazon DocumentDB darüber informiert, wie das Feld „Koordinaten“ interpretiert wird. Derzeit unterstützt Amazon DocumentDB nur Punkte
- Koordinaten – ein Breiten- und Längengradpaar, das als Objekt in einem Array dargestellt wird – [Länge, Breitengrad]

Amazon DocumentDB verwendet auch 2dSphere-Indizes, um Geodaten zu indizieren. Amazon DocumentDB unterstützt Indizierungspunkte. Amazon DocumentDB unterstützt Näherungsabfragen mit 2dSphere-Indizierung.

Betrachten wir ein Szenario, in dem Sie eine Anwendung für den Lebensmittelzustellungsdienst erstellen. Sie möchten die Breiten- und Längengradpaare verschiedener Studios in Amazon DocumentDB speichern. Dazu empfehlen wir, zunächst einen Index im Geodatenfeld zu erstellen, das das Breiten- und Längengradpaar enthält.

```
use restaurantsdb
db.usarestaurants.createIndex({location:"2dsphere"})
```

Die Ausgabe dieses Befehls würde etwa wie folgt aussehen:

```
{
 "createdCollectionAutomatically" : true,
 "numIndexesBefore" : 1,
 "numIndexesAfter" : 2,
 "ok" : 1
}
```

Sobald Sie einen Index erstellt haben, können Sie mit dem Einfügen von Daten in Ihre Amazon DocumentDB-Sammlung beginnen.

```
db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Thai Palace",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
         -122.3264,
         47.6009
      ]
   }
});

db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Noodle House",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
        -122.3517,
         47.6159
      ]
   }
});

db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Curry House",
   "rating": 4.8,
```

```
   "location":{
      "type":"Point",
      "coordinates":[
         -121.4517,
         47.6229
      ]
   }
});
```

Abfragen von koordinatenbasierten Daten

Amazon DocumentDB unterstützt Nähe, Aufnahme und Schnittpunktabfrage von Geodaten. Ein gutes Beispiel für eine Näherungsabfrage ist das Auffinden aller Punkte (alle Flughafen), die kleiner als eine bestimmte Entfernung und mehr als eine Entfernung von einem anderen Punkt (Stadt) sind. Ein gutes Beispiel für die Einschlussabfrage ist, alle Punkte (alle Flughafen) zu finden, die sich in einem bestimmten Gebiet/Polygon (Bundesstaat New York) befinden. Ein gutes Beispiel für eine Schnittpunktabfrage ist die Suche nach einem Polygon (Zustand), das sich mit einem Punkt (Stadt) schneidet. Sie können die folgenden Geodatenoperatoren verwenden, um Erkenntnisse aus Ihren Daten zu gewinnen.

- **`$nearSphere`** – `$nearSphere` ist ein Suchoperator, der das Suchen von Punkten von am nächsten zu am weitesten von einem GeoJSON-Punkt unterstützt.
- **`$geoNear`** – `$geoNear` ist ein Aggregationsoperator, der die Berechnung der Entfernung in Metern von einem GeoJSON-Punkt unterstützt.
- **`$minDistance`** – `$minDistance` ist ein Suchoperator, der in Verbindung mit `$nearSphere` oder verwendet wird, `$geoNear` um Dokumente zu filtern, die mindestens den angegebenen Mindestabstand zum Mittelpunkt aufweisen.
- **`$maxDistance`** – `$maxDistance` ist ein Suchoperator, der in Verbindung mit `$nearSphere` oder verwendet wird, `$geoNear` um Dokumente zu filtern, die maximal die angegebene maximale Entfernung vom Mittelpunkt erreichen.
- **`$geoWithin`** – `$geoWithin` ist ein Suchoperator, der das Suchen von Dokumenten mit Geodaten unterstützt, die vollständig innerhalb einer bestimmten Form wie einem Polygon existieren.
- **`$geoIntersects`** – `$geoIntersects` ist ein Suchoperator, der das Suchen von Dokumenten unterstützt, deren Geodaten sich mit einem angegebenen GeoJSON-Objekt schneiden.

> **Note**
>
> `$geoNear` und `$nearSphere` erfordern einen 2dSphere-Index für das GeoJSON-Feld, das Sie in Ihrer Näherungsabfrage verwenden.

Beispiel 1

In diesem Beispiel erfahren Sie, wie Sie alle Ker (Punkt) sortiert nach der nächstgelegenen Entfernung von einer Adresse (Punkt) finden.

Um eine solche Abfrage durchzuführen, können Sie verwenden, `$geoNear` um die Entfernung von Punkten von einem anderen Punkt zu berechnen. Sie können auch hinzufügen`distanceMultiplier`, um die Entfernung in Kilometern zu messen.

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001
      }
   }
])
```

Der obige Befehl würde Trichter zurückgeben, sortiert nach Entfernung (am nächsten zu am weitesten von dem angegebenen Punkt). Die Ausgabe dieses Befehls würde ungefähr so aussehen

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "state" : "Washington", "city" :
 "Seattle", "name" : "Noodle House", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -122.3517, 47.6159 ] }, "DistanceKilometers" : 0.03422834547294996 }
{ "_id" : ObjectId("611f3da185009a81ad38e74a"), "state" : "Washington", "city" :
 "Seattle", "name" : "Thai Palace", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -122.3264, 47.6009 ] }, "DistanceKilometers" : 2.5009390081704277 }
```

```
{ "_id" : ObjectId("611f3dae85009a81ad38e74c"), "state" : "Washington", "city" :
 "Seattle", "name" : "Curry House", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -121.4517, 47.6229 ] }, "DistanceKilometers" : 67.52845344856914 }
```

Um die Anzahl der Ergebnisse in einer Abfrage zu begrenzen, verwenden Sie die num Option `limit` oder .

`limit`:

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001,
         "limit": 10
      }
   }
])
```

num:

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001,
```

```
         "num": 10
      }
   }
])
```

> **Note**
>
> `$geoNear` stage unterstützt die `num` Optionen `limit` und , um die maximale Anzahl zurückzugebender Dokumente anzugeben. `$geoNear` gibt standardmäßig maximal 100 Dokumente zurück, wenn die `num` Optionen `limit` oder nicht angegeben sind. Dies wird durch den Wert der `$limit` Stufe überschrieben, falls vorhanden, und der Wert ist kleiner als 100.

Beispiel 2

In diesem Beispiel erfahren Sie, wie Sie alle Ker (Punkt) innerhalb von 2 Kilometern von einer bestimmten Adresse (Punkt) finden. Um eine solche Abfrage auszuführen, können Sie `$nearSphere` innerhalb eines Minimums `$minDistance` und Maximums von einem GeoJSON-Punkt `$maxDistance` aus verwenden

```
db.usarestaurants.find({
   "location":{
      "$nearSphere":{
         "$geometry":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "$minDistance":1,
         "$maxDistance":2000
      }
   }
},
{
   "name":1
})
```

Der obige Befehl würde Ker in einer maximalen Entfernung von 2 Kilometern vom angegebenen Punkt zurückgeben. Die Ausgabe dieses Befehls würde ungefähr so aussehen

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "name" : "Noodle House" }
```

Einschränkungen

Amazon DocumentDB unterstützt keine Abfrage oder Indizierung von Polygonen, LineString MultiPoint MultiPolygon MultiLineString, und GeometryCollection.

Teilweiser Index

Ein Teil indiziert Dokumente in einer Sammlung, die ein bestimmtes Filterkriterium erfüllt. Die teilweise Indexfunktion wird in Instance-basierten Clustern von Amazon DocumentDB 5.0 unterstützt.

Themen

- Erstellen eines Teilindex
- Unterstützte Operatoren
- Abfragen mit einem Teilindex
- Funktionen für partielle Indizes
- Einschränkungen des Teilindex

Erstellen eines Teilindex

Um einen partiellen Index zu erstellen, verwenden Sie die -`createIndex()`Methode mit der -`partialFilterExpression`Option. Die folgende Operation erstellt beispielsweise einen eindeutigen zusammengesetzten Index in der Auftragssammlung, der Dokumente indiziert, die einen `OrderID` und das `isDelivered` Feld als wahr haben:

```
db.orders.createIndex(
  {"category": 1, "CustomerId": 1, "OrderId": 1},
  {"unique": true, "partialFilterExpression":
    {"$and": [
      {"OrderId": {"$exists": true}},
      {"isDelivered": {"$eq": false}}
    ]}
  }
```

```
)
```

Unterstützte Operatoren

- $eq
- $exists
- $and (nur auf oberster Ebene)
- $gt/$gte/$lt/$lte (Indexscan wird nur verwendet, wenn der in der Abfrage angegebene Filter genau mit dem teilweisen Filterausdruck übereinstimmt) (siehe Einschränkungen)

Abfragen mit einem Teilindex

Die folgenden Abfragemuster sind mit Teilindizes möglich:

- Das Abfrageprädikat stimmt genau mit dem partiellen Indexfilterausdruck überein:

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}}
  ]}).explain()
```

- Das erwartete Ergebnis des Abfragefilters ist eine logische Teilmenge des Teilfilters:

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}},
    {"OrderAmount": {"$eq": "5"}}
  ]}).explain()
```

- Ein Unterprädikat der Abfrage kann in Verbindung mit anderen Indizes verwendet werden:

```
db.orders.createIndex({"anotherIndex":1})
db.orders.find({ "$or": [
      {"$and": [
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]},
      {"anotherIndex": {"$eq": 5}}
    ]
```

```
}).explain()
```

> **Note**
>
> Ein Abfrageplaner kann sich dafür entscheiden, einen Sammlungs-Scan anstelle eines Index-Scans zu verwenden, wenn dies effizient ist. Dies wird in der Regel bei sehr kleinen Sammlungen oder Abfragen beobachtet, die einen großen Teil einer Sammlung zurückgeben würden.

Funktionen für partielle Indizes

Teilindizes auflisten

Listen Sie Teilindizes mit partialFilterExpression mithilfe der -`getIndex`Operation auf. Die in ausgegebene `getIndex` Operation listet beispielsweise Teilindizes mit den Feldern key, name und partialfilterExpressions auf:

```
db.orders.getIndexes()
```

In diesem Beispiel wird die folgende Ausgabe zurückgegeben:

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
  },
  {
    "v" : 4,
    "unique" : true,
    "key" : {
      "category" : 1,
      "" : 1,
      "CustomerId" : 1,
      "OrderId" : 1
    },
```

```
    "name" : "category_1_CustID_1_OrderId_1",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {
      "$and" : [
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]
    }
  }
]
```

Mehrere partielle Filterausdrücke für denselben Schlüssel:Reihenfolge

Für dieselben Feldkombinationen (key:order) können unterschiedliche Teilindizes erstellt werden. Diese Indizes müssen einen anderen Namen haben.

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"firstPartialIndex",
    partialFilterExpression:{"OrderId":{"$exists": true}}
  }
)
```

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"secondPartialIndex",
    partialFilterExpression:{"OrderId":{"$gt": 1000}}
  }
)
```

Führen Sie den `getIndexes` Vorgang aus, um alle Indizes in der Sammlung aufzulisten:

```
db.orders.getIndexes()
```

Diese Beispiele geben die folgende Ausgabe zurück:

```
[
  {
    "v" : 4,
    "key" : {
```

```
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
  },
  {
    "v" : 4,
    "key" : {
      "OrderId" : 1
    },
    "name" : "firstPartialIndex",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {"OrderId":{"$exists": true}}
  },
  {
    "v" : 4,
    "key" : {
      "OrderId" : 1
    },
    "name" : "secondPartialIndex",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {"OrderId":{"$gt": 1000}}
  }
]
```

> **Important**
>
> Indexnamen müssen unterschiedlich sein und dürfen nur nach Namen gelöscht werden.

Indizes mit Teil- und TTL-Eigenschaften

Sie können auch Indizes mit Teil- und TTL-Eigenschaften erstellen, indem Sie während der Indexerstellung die `expireAfterSeconds` Optionen `partialFilterExpression` und angeben. Auf diese Weise können Sie besser steuern, welche Dokumente jetzt aus einer Sammlung entfernt werden.

Sie können beispielsweise einen TTL-Index haben, der Dokumente identifiziert, die nach einem bestimmten Zeitraum gelöscht werden sollen. Sie können jetzt zusätzliche Bedingungen dafür angeben, wann Dokumente mit der Option Teilindex gelöscht werden sollen:

```
db.orders.createIndex(
```

```
    { "OrderTimestamp": 1 },
    {
        expireAfterSeconds: 3600 ,
        partialFilterExpression: { "isDelivered": { $eq: true } }
    }
)
```

In diesem Beispiel wird die folgende Ausgabe zurückgegeben:

```
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1,
        "operationTime" : Timestamp(1234567890, 1)
}
```

Führen Sie die -`getIndexes`Operation aus, um Indizes aufzulisten, die in der Sammlung vorhanden sind:

```
db.orders.getIndexes()
[
    {
        "v" : 4,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.orders"
    }
```

In diesem Beispiel wird die folgende Ausgabe zurückgegeben:

```
[
    {
        "v": 4,
        "key": {
            "_id": 1
        },
        "name": "_id_",
        "ns": "ecommerceApp.orders"
    },
```

```
    {
        "v": 4,
        "key": {
            "OrderTimestamp": 1
        },
        "name": "OrderTimestamp_1",
        "ns": "ecommerceApp.orders",
        "partialFilterExpression": {
            "isDelivered": {
                "$eq": true
            }
        },
        "expireAfterSeconds": 3600
    }
]
```

Einschränkungen des Teilindex

Die folgenden Einschränkungen gelten für die Teilindexfunktion:

- Ungleichheitsabfragen in Amazon DocumentDB verwenden nur dann einen partiellen Index, wenn das Abfragefilterprädikat genau mit dem übereinstimmt `partialFilterExpression` und vom gleichen Datentyp ist.

  > **Note**
  >
  > Auch kann für den obigen Fall `$hint` nicht verwendet werden, um IXSCAN zu erzwingen.

  Im folgenden Beispiel `partialFilterExpression` wird die nur auf angewendet, `field1` aber nicht auf `field2`:

```
db.orders.createIndex(
  {"OrderAmount": 1},
  {"partialFilterExpression": { OrderAmount : {"$gt" : 5}}}
)

db.orders.find({OrderAmount : {"$gt" : 5}}) // Will use partial index
db.orders.find({OrderAmount : {"$gt" : 6}}) // Will not use partial index
db.orders.find({OrderAmount : {"$gt" : Decimal128(5.00)}}) // Will not use partial
 index
```

- Ein `partialFilterExpression` mit Array-Operatoren wird nicht unterstützt. Der folgende Vorgang generiert einen Fehler:

```
db.orders.createIndex(
  {"CustomerId":1},
  {'partialFilterExpression': {'OrderId': {'$eq': [1000, 1001, 1002]}}}
)
```

- Die folgenden Operatoren werden im partialFilterExpression Feld nicht unterstützt:
  - `$all` (Array-Operator)
  - `$mod` (Array-Operator)
  - `$or`
  - `$xor`
  - `$not`
  - `$nor`
- Der Datentyp des Filterausdrucks und der Filter sollten identisch sein.

Durchführen einer Textsuche mit Amazon DocumentDB

Mit der nativen Volltextsuchfunktion von Amazon DocumentDB können Sie Textsuchen für große Textdatensätze mithilfe von speziellen Textindizes durchführen. In diesem Abschnitt werden die Funktionen der Textindexfunktion beschrieben und Schritte zum Erstellen und Verwenden von Textindizes in Amazon DocumentDB bereitgestellt. Einschränkungen bei der Textsuche werden ebenfalls aufgeführt.

Themen

Unterstützte Funktionen

Die Amazon DocumentDB-Textsuche unterstützt die folgenden mit der MongoDB-API kompatiblen Funktionen:

- Erstellen Sie Textindizes für ein einzelnes Feld.
- Erstellen Sie zusammengesetzte Textindizes, die mehr als ein Textfeld enthalten.
- Führen Sie Einzelwort- oder Mehrwortsuchen durch.
- Steuern Sie die Suchergebnisse mithilfe von Gewichtungen.
- Sortieren Sie die Suchergebnisse nach Punktzahl.
- Verwenden Sie den Textindex in der Aggregationspipeline.
- Suchen Sie nach einer exakten Phrase.

Verwenden des Amazon DocumentDB-Textindex

Um einen Textindex für ein Feld zu erstellen, das Zeichenfolgendaten enthält, geben Sie die Zeichenfolge „Text“ an, wie unten gezeigt:

Einzelfeldindex:

```
db.test.createIndex({"comments": "text"})
```

Dieser Index unterstützt Textsuchabfragen im Zeichenfolgenfeld „Kommentare“ in der angegebenen Sammlung.

Erstellen Sie einen zusammengesetzten Textindex für mehr als ein Zeichenfolgenfeld:

```
db.test.createIndex({"comments": "text", "title":"text"})
```

Dieser Index unterstützt Textsuchabfragen in den Zeichenfolgenfeldern „Kommentare“ und „Titel“ in der angegebenen Sammlung. Sie können beim Erstellen eines zusammengesetzten Textindexes bis zu 30 Felder angeben. Nach der Erstellung fragen Ihre Textsuchabfragen alle indizierten Felder ab.

> **Note**
>
> Für jede Sammlung ist nur ein Textindex zulässig.

Auflisten eines Textindex in einer Amazon DocumentDB-Sammlung

Sie können `getIndexes()` für Ihre Sammlung verwenden, um Indizes zu identifizieren und zu beschreiben, einschließlich Textindizes, wie im folgenden Beispiel gezeigt:

```
rs0:PRIMARY> db.test.getIndexes()
[
   {
      "v" : 4,
      "key" : {
         "_id" : 1
      },
      "name" : "_id_",
      "ns" : "test.test"
   },
   {
      "v" : 1,
      "key" : {
         "_fts" : "text",
         "_ftsx" : 1
      },
      "name" : "contents_text",
      "ns" : "test.test",
      "default_language" : "english",
      "weights" : {
         "comments" : 1
      },
      "textIndexVersion" : 1
   }
]
```

Sobald Sie einen Index erstellt haben, beginnen Sie mit dem Einfügen von Daten in Ihre Amazon DocumentDB-Sammlung.

```
db.test.insertMany([{"_id": 1, "star_rating": 4, "comments": "apple is red"},
                    {"_id": 2, "star_rating": 5, "comments": "pie is delicious"},
                    {"_id": 3, "star_rating": 3, "comments": "apples, oranges - healthy
 fruit"},
                    {"_id": 4, "star_rating": 2, "comments": "bake the apple pie in the
 oven"},
                    {"_id": 5, "star_rating": 5, "comments": "interesting couch"},
```

```
                    {"_id": 6, "star_rating": 5, "comments": "interested in couch for
 sale, year 2022"}])
```

Ausführen von Textsuchabfragen

Ausführen einer Einzelwort-Textsuchabfrage

Sie müssen die `$search` Operatoren `$text` und verwenden, um Textsuchen durchzuführen. Das folgende Beispiel gibt alle Dokumente zurück, in denen Ihr textindiziertes Feld die Zeichenfolge „apple“ oder „apple“ in anderen Formaten wie „apples“ enthält:

```
db.test.find({$text: {$search: "apple"}})
```

Ausgabe:

Die Ausgabe dieses Befehls sieht etwa wie folgt aus:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Ausführen einer Textsuche mit mehreren Wörtern

Sie können auch Textsuchen mit mehreren Wörtern für Ihre Amazon DocumentDB-Daten durchführen. Der folgende Befehl gibt Dokumente mit einem textindizierten Feld zurück, das „apple“ oder „pie“ enthält:

```
db.test.find({$text: {$search: "apple pie"}})
```

Ausgabe:

Die Ausgabe dieses Befehls sieht etwa wie folgt aus:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Ausführen einer Textsuche mit mehreren Wörtern

Verwenden Sie dieses Beispiel für eine Wortgruppensuche:

```
db.test.find({$text: {$search: "\"apple pie\""}})
```

Ausgabe:

Der obige Befehl gibt Dokumente mit einem textindizierten Feld zurück, das den genauen Satz „apple pie“ enthält. Die Ausgabe dieses Befehls sieht etwa wie folgt aus:

```
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Ausführen einer Textsuche mit Filtern

Sie können die Textsuche auch mit anderen Abfrageoperatoren kombinieren, um Ergebnisse basierend auf zusätzlichen Kriterien zu filtern:

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "interest"}}]})
```

Ausgabe:

Der obige Befehl gibt Dokumente mit einem textindizierten Feld zurück, das jede Form von „interest“ und „star_rating“ gleich 5 enthält. Die Ausgabe dieses Befehls sieht etwa wie folgt aus:

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
{ "_id" : 6, "star_rating" : 5, "comments" : "interested in couch for sale, year
 2022" }
```

Begrenzen der Anzahl der Dokumente, die bei einer Textsuche zurückgegeben werden

Sie können die Anzahl der zurückgegebenen Dokumente mithilfe von einschränken`limit`:

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "couch"}}]}).limit(1)
```

Ausgabe:

Der obige Befehl gibt ein Ergebnis zurück, das den Filter erfüllt:

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
```

Ergebnisse nach Textbewertung sortieren

Im folgenden Beispiel werden die Textsuchergebnisse nach Textbewertung sortiert:

```
db.test.find({$text: {$search: "apple"}}, {score: {$meta: "textScore"}}).sort({score:
 {$meta: "textScore"}})
```

Ausgabe:

Der obige Befehl gibt Dokumente mit einem textindizierten Feld zurück, das „apple“ oder „apple“ in anderen Formaten wie „apples“ enthält, und sortiert das Ergebnis basierend darauf, wie relevant das Dokument mit dem Suchbegriff zusammenhängt. Die Ausgabe dieses Befehls sieht etwa wie folgt aus:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red", "score" :
 0.6079270860936958 }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit",
 "score" : 0.6079270860936958 }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven",
 "score" : 0.6079270860936958 }
```

`$text` und `$search` werden auch für die `delete` Befehle `aggregate`, `count`, `findAndModifyupdate`, und unterstützt.

Aggregationsoperatoren

Aggregationspipeline mit **`$match`**

```
db.test.aggregate(
   [{ $match: { $text: { $search: "apple pie" } } }]
)
```

Ausgabe:

Der obige Befehl gibt die folgenden Ergebnisse zurück:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apple - a healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
```

Eine Kombination anderer Aggregationsoperatoren

```
db.test.aggregate(
   [
```

```
      { $match: { $text: { $search: "apple pie" } } },
      { $sort: { score: { $meta: "textScore" } } },
      { $project: { score: { $meta: "textScore" } } }
   ]
)
```

Ausgabe:

Der obige Befehl gibt die folgenden Ergebnisse zurück:

```
{ "_id" : 4, "score" : 0.6079270860936958 }
{ "_id" : 1, "score" : 0.3039635430468479 }
{ "_id" : 2, "score" : 0.3039635430468479 }
{ "_id" : 3, "score" : 0.3039635430468479 }
```

Geben Sie beim Erstellen eines Textindex mehrere Felder an

Sie können bis zu drei Feldern in Ihrem zusammengesetzten Textindex Gewichtungen zuweisen. Die einem Feld in einem Textindex zugewiesene Standardgewichtung ist eins (1). Die Gewichtung ist ein optionaler Parameter und muss im Bereich von 1 bis 10 000 liegen.

```
db.test.createIndex(
   {
     "firstname": "text",
     "lastname": "text",
     ...
   },
   {
     weights: {
       "firstname": 5,
       "lastname":10,
       ...
     },
     name: "name_text_index"
   }
 )
```

Unterschiede zu MongoDB

Das Textindexfeature von Amazon DocumentDB verwendet einen invertierten Index mit einem Begriffsfrequenzalgorithmus. Textindizes sind standardmäßig spärlich. Aufgrund von Unterschieden

in der Parsing-Logik, den Trennzeichen für die Tokenisierung und anderen wird dieselbe Ergebnismenge wie MongoDB möglicherweise nicht für denselben Datensatz oder dieselbe Abfrageform zurückgegeben.

Es gibt die folgenden zusätzlichen Unterschiede zwischen dem Amazon DocumentDB-Textindex und MongoDB:

- Zusammengesetzte Indizes, die Nicht-Textindizes verwenden, werden nicht unterstützt.
- Bei Amazon DocumentDB-Textindizes wird die Groß- und Kleinschreibung nicht beachtet und bei Diakritik wird nicht berücksichtigt.
- Nur Englisch wird mit dem Textindex unterstützt.
- Die Textindizierung von Array-Feldern (oder Feldern mit mehreren Schlüsseln) wird nicht unterstützt. Beispielsweise schlägt das Erstellen eines Textindex auf „a“ mit dem Dokument {“a“: [“apple“, „pie“]} fehl.
- Die Platzhaltertextindizierung wird nicht unterstützt.
- Eindeutige Textindizes werden nicht unterstützt.
- Das Ausschließen eines Begriffs wird nicht unterstützt.

Bewährte Methoden und Richtlinien

- Für eine optimale Leistung bei Textsuchabfragen, bei denen nach Textwerten sortiert wird, empfehlen wir Ihnen, den Textindex zu erstellen, bevor Sie Daten laden.
- Textindizes erfordern zusätzlichen Speicher für eine optimierte interne Kopie der indizierten Daten. Dies hat zusätzliche Auswirkungen auf die Kosten.

Einschränkungen

Die Textsuche hat in Amazon DocumentDB die folgenden Einschränkungen:

- Die Textsuche wird nur auf Instance-basierten Clustern von Amazon DocumentDB 5.0 unterstützt.

Verbindungsprobleme Amazon DocumentDB

Die folgenden Abschnitte enthalten Informationen zur Behebung von Problemen, die bei der Verwendung von Amazon DocumentDB (mit MongoDB-Kompatibilität) auftreten können.

Themen

Verbindungsprobleme bei Verbindungen

Haben Sie Probleme mit der Verbindung? Hier sind einige gängige Szenarien und wie diese gelöst werden können.

Topics

Es kann keine Verbindung zu einem Amazon DocumentDB DocumentDB-Endpunkt hergestellt werden

Wenn Sie versuchen, eine Verbindung zu Amazon DocumentDB herzustellen, wird im Folgenden eine der häufigsten Fehlermeldungen angezeigt, die Sie möglicherweise erhalten.

```
connecting to: mongodb://docdb-2018-11-08-21-47-27.cluster-ccuszbx3pn5e.us-east-
1.docdb.amazonaws.com:27017/
2018-11-14T14:33:46.451-0800 W NETWORK [thread1] Failed to connect to
172.31.91.193:27017 after 5000ms milliseconds, giving up.
2018-11-14T14:33:46.452-0800 E QUERY [thread1] Error: couldn't connect to server
docdb-2018-11-08-21-47-27.cluster-ccuszbx3pn5e.us-east-1.docdb.amazonaws.com:27017,
connection attempt failed :
connect@src/mongo/shell/mongo.js:237:13
```

```
@(connect):1:6
exception: connect failed
```

Diese Fehlermeldung bedeutet normalerweise, dass Ihr Client (die Mongo-Shell in diesem Beispiel) nicht auf den Amazon DocumentDB DocumentDB-Endpunkt zugreifen kann. Dies kann aus verschiedenen Gründen der Fall sein:

Themen

- Verbindung von öffentlichen Endpunkten aus herstellen
- Regionsübergreifende Verbindungen
- Verbindung von verschiedenen Amazon-VPCs aus herstellen
- Sicherheitsgruppe blockiert eingehende Verbindungen
- Problem mit den Leseeinstellungen des Java Mongo-Treibers

Verbindung von öffentlichen Endpunkten aus herstellen

Sie versuchen, direkt von Ihrem Laptop oder Ihrem lokalen Entwicklungscomputer aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen.

Der Versuch, direkt von einem öffentlichen Endpunkt aus, wie Ihrem Laptop oder Ihrem lokalen Entwicklungscomputer, eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, schlägt fehl. Amazon DocumentDB ist nur für Virtual Private Cloud (VPC) verfügbar und unterstützt derzeit keine öffentlichen Endgeräte. Daher können Sie von Ihrem Laptop oder Ihrer lokalen Entwicklungsumgebung außerhalb Ihrer VPC keine direkte Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Um von außerhalb einer Amazon VPC eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, können Sie einen SSH-Tunnel verwenden. Weitere Informationen finden Sie unter Verbindung zu einem Amazon DocumentDB-Cluster von außerhalb einer Amazon VPC herstellen. Wenn sich Ihre Entwicklungsumgebung in einer anderen Amazon VPC befindet, können Sie außerdem VPC Peering verwenden und von einer anderen Amazon VPC in derselben Region oder einer anderen Region aus eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Regionsübergreifende Verbindungen

Sie versuchen, eine Verbindung zu einem Amazon DocumentDB-Cluster in einer anderen Region herzustellen.

Wenn Sie versuchen, eine Verbindung mit einem Amazon-DocumentDB-Cluster von einer Amazon-EC2-Instance in einer anderen Region USA Ost (Nord-Virginia) (us-east-1) aus Region USA Ost (Nord-Virginia) (us-west-2) mit einem Cluster in Region USA Ost (Nord-Virginia) (us-west-2) mit einem Cluster in Region USA Ost (Nord-Virginia) (us-east-1) mit Region USA Ost (Nord-Virginia) (us-east-1) mit Region USA Ost (Nord-Virginia) (us-east-1) mit einem Cluster in Region USA Ost (Nord-Virginia) (us-east-1

Führen Sie den folgenden Befehl aus, um die Region Ihres Amazon-DocumentDB-Clusters zu überprüfen. Die Region ist im Endpunkt.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].Endpoint'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    "sample-cluster.node.us-east-1.docdb.amazonaws.com"
]
```

Um die Region Ihrer EC2-Instance zu überprüfen, führen Sie den folgenden Befehl aus.

```
 aws ec2 describe-instances \
     --query 'Reservations[*].Instances[*].Placement.AvailabilityZone'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "us-east-1a"
    ]
]
```

Verbindung von verschiedenen Amazon-VPCs aus herstellen

Sie versuchen, eine Verbindung zu einem Amazon DocumentDB-Cluster von einer VPC aus herzustellen, die sich von der Amazon VPC unterscheidet, in der Ihr Cluster bereitgestellt wird.

Wenn sich sowohl Ihr Amazon DocumentDB-Cluster als auch Ihre Amazon EC2 EC2-Instance in derselbenAWS-Region, aber nicht in derselben Amazon VPC befinden, können Sie keine direkte

Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen, es sei denn, VPC Peering ist zwischen den beiden Amazon VPCs aktiviert.

Führen Sie den folgenden Befehl aus, um die Amazon-VPC Ihrer Amazon-DocumentDB-Instance zu überprüfen.

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

Führen Sie den folgenden Befehl aus, um die Amazon-VPC Ihrer Amazon-EC2-Instance zu überprüfen.

```
aws ec2 describe-instances \
   --query 'Reservations[*].Instances[*].VpcId'
```

Sicherheitsgruppe blockiert eingehende Verbindungen

Sie versuchen, eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, und die Sicherheitsgruppe des Clusters erlaubt keine eingehenden Verbindungen auf dem Port des Clusters (Standardport: 27017).

Angenommen, Ihr Amazon DocumentDB-Cluster und Ihre Amazon EC2 EC2-Instance befinden sich beide in derselben Region und Amazon VPC und verwenden dieselbe Amazon VPC-Sicherheitsgruppe. Wenn Sie keine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen können, liegt die wahrscheinliche Ursache darin, dass Ihre Sicherheitsgruppe (d. h. die Firewall) für Ihren Cluster keine eingehenden Verbindungen an dem Port zulässt, den Sie für Ihren Amazon DocumentDB-Cluster ausgewählt haben (Standardport ist 27017).

Führen Sie den folgenden Befehl aus, um den Port für Ihren Amazon-DocumentDB-Cluster zu überprüfen.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port]'
```

Führen Sie den folgenden Befehl aus, um die Amazon-DocumentDB-Sicherheitsgruppe für den Cluster abzurufen.

```
aws docdb describe-db-clusters \
```

```
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[VpcSecurityGroups[*],VpcSecurityGroupId]'
```

Informationen zur Überprüfung der Regeln für eingehenden Datenverkehr für Ihre Sicherheitsgruppe finden Sie in den folgenden Themen in der Amazon-EC2-Dokumentation:

- Autorisieren von eingehendem Traffic für Ihre Linux-Instances
- Autorisieren von eingehendem Datenverkehr für Ihre Windows-Instances

Problem mit den Leseeinstellungen des Java Mongo-Treibers

Die Leseeinstellungen der Clients werden nicht berücksichtigt, und einige Clients können nach einem Failover nicht in Amazon DocumentDB schreiben, es sei denn, sie starten neu.

Dieses Problem, das erstmals in Java Mongo Driver 3.7.x entdeckt wurde, tritt auf, wenn ein Client mithilfe `MongoClientSettings` und insbesondere beim Verketten der Methode eine Verbindung zu Amazon DocumentDB herstellt. `applyToClusterSettings` Die MongoClient Clustereinstellungen können mit verschiedenen Methoden definiert werden, z. B. `hosts()requiredReplicaSetName()`, und`mode()`.

Wenn der Client in der `hosts()` Methode nur einen Host angibt, wird der Modus auf `ClusterConnectionMode.SINGLE` statt auf gesetzt. `ClusterConnectionMode.MULTIPLE` Dies führt dazu, dass der Client die Lesepräferenz ignoriert und nur eine Verbindung zu dem in `hosts()` konfigurierten Server herstellt. Selbst wenn die Client-Einstellungen wie unten initialisiert werden, würden alle Lesevorgänge immer noch an die primäre statt an die sekundäre gehen.

```
final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
    final MongoCredential credential = MongoCredential.createCredential("xxx",
            "admin", "xxxx".toCharArray());
    final MongoClientSettings settings = MongoClientSettings.builder()
            .credential(credential)
            .readPreference(ReadPreference.secondaryPreferred())
            .retryWrites(false)
            .applyToSslSettings(builder -> builder
                    .enabled(false))
            .applyToClusterSettings(builder -> builder.hosts(
                            Arrays.asList(serverAddress0
                            ))
                    .requiredReplicaSetName("rs0"))
            .build();
```

```
    MongoClient mongoClient = MongoClients.create(settings);
```

Failover-Koffer

Bei Verwendung der oben genannten Client-Verbindungseinstellungen würde der Client im Falle eines Failovers und einer verzögerten Aktualisierung des DNS-Eintrags für den Cluster-Writer-Endpunkt immer noch versuchen, Schreibvorgänge an den alten Writer zu senden (nach einem Failover jetzt Reader). Dies führt zu einem serverseitigen Fehler (nicht Master), der vom Java-Treiber nicht angemessen behandelt wird (dies wird noch untersucht). So kann der Client beispielsweise so lange in einem schlechten Zustand bleiben, bis der Anwendungsserver neu gestartet wird.

Dafür gibt es zwei Workarounds:

- Bei Clients, die über eine Verbindungszeichenfolge eine Verbindung zu Amazon DocumentDB herstellen, tritt dieses Problem nicht auf, da `ClusterConnectionMode` es `MULTIPLE` bei der Einstellung der Lesepräferenz auf gesetzt wird.

  ```
  MongoClientURI mongoClientURI = new MongoClientURI("mongodb://usr:pass:cluster-
  endpoint:27317/test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred");
  MongoClient mongoClient = MongoClients.create(mongoClientURI.getURI());
  ```

  Oder verwenden Sie `MongoClientSettings` Builder mit der `applyConnectionString` Methode.

  ```
  final MongoClientSettings settings = MongoClientSettings.builder()
          .credential(credential)
          .applyConnectionString(new ConnectionString("usr:pass:cluster-endpoint:27317/
  test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred"))
          .retryWrites(false)
          .applyToSslSettings(builder # builder
                  .enabled(false))
          .build();
  MongoClient mongoClient = MongoClients.create(settings);
  ```

- Explizit `ClusterConnectionMode` auf gesetzt`MULTIPLE`. Dies ist nur erforderlich, wenn Sie `applyToClusterSettings` und verwenden`hosts().size() == 1`.

  ```
  final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
  final MongoCredential credential = MongoCredential.createCredential("xxx","admin",
   "xxxx".toCharArray());
  final MongoClientSettings settings = MongoClientSettings.builder()
  ```

```
    .credential(credential)
    .readPreference(ReadPreference.secondaryPreferred())
    .retryWrites(false)
    .applyToSslSettings(builder # builder
    .enabled(false))
    .applyToClusterSettings(builder # builder
                .hosts(Arrays.asList(serverAddress0))
                .requiredReplicaSetName("rs0"))
                .mode(ClusterConnectionMode.MULTIPLE))
    .build();
MongoClient mongoClient = MongoClients.create(settings);
```

Testen der Verbindung zu einer Amazon-DocumentDB-Instance

Sie können Ihre Verbindung zu einem Cluster mit gängigen Linux- oder Windows-Tools testen.

Testen Sie die Verbindung über einen Linux- oder Unix-Terminal, indem Sie folgendes eingeben (ersetzen Sie `cluster-endpoint` durch den Endpunkt und ersetzen Sie `port` durch den Port Ihrer Instance).

```
nc -zv cluster-endpoint port
```

Der folgende Code ist ein Beispiel für eine Beispieloperation und den Rückgabewert:

```
nc -zv docdbTest.d4c7nm7stsfc0.us-west-2.docdb.amazonaws.com 27017

Connection to docdbTest.d4c7nm7stsfc0.us-west-2.docdb.amazonaws.com 27017 port [tcp/*]
 succeeded!
```

Verbindung zu einem ungültigen Endpunkt herstellen

Wenn Sie eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen und Sie einen Cluster-Endpunkt verwenden, der nicht gültig ist, wird ein Fehler ähnlich dem folgenden angezeigt.

```
mongo --ssl \
   --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 \
   --sslCAFile global-bundle.pem \
   --username <user-name> \
   --password <password>
```

Das Ergebnis sieht folgendermaßen aus:

```
MongoDB shell version v3.6
connecting to: mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
2018-11-14T17:21:18.516-0800 I NETWORK [thread1] getaddrinfo("sample-cluster.node.us-
east-1.docdb.amazonaws.com") failed:
nodename nor servname provided, or not known 2018-11-14T17:21:18.537-0800 E QUERY
 [thread1] Error: couldn't initialize
connection to host sample-cluster.node.us-east-1.docdb.amazonaws.com, address is
 invalid :
connect@src/mongo/shell/mongo.js:237:13@(connect):1:6
exception: connect failed
```

Um den gültigen Endpunkt für einen Cluster abzurufen, führen Sie den folgenden Befehl aus:

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[Endpoint,Port]'
```

Um den gültigen Endpunkt für eine Instance abzurufen, führen Sie den folgenden Befehl aus:

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].[Endpoint.Address,Endpoint.Port]'
```

Weitere Informationen finden Sie unter Grundlegendes zu Amazon DocumentDB-Endpunkten.

Indexerstellung bei Indexerstellung

Die folgenden Themen erläutern, was zu tun ist, wenn Ihr Index oder Ihr Indexaufbau im Hintergrund fehlschlägt.

Themen

- Index-Erstellung schlägt fehl
- Latenzprobleme bei der Erstellung des Hintergrundindexes und Fehler

Index-Erstellung schlägt fehl

Amazon DocumentDB verwendet im Rahmen der Indexerstellung lokalen Speicher auf einer Instance. Sie können diese Festplattennutzung mithilfe der FreeLocalStorageCloudWatchMetrik (`CloudWatch -> Metrics -> DocDB -> Instance Metrics`) überwachen. Wenn ein Indexaufbau den gesamten lokalen Festplattenspeicherplatz verbraucht und fehlschlägt, erhalten Sie eine Fehlermeldung. Bei der Migration von Daten zu Amazon DocumentDB empfehlen wir Ihnen, zuerst Indizes zu erstellen und dann die Daten einzufügen. Weitere Informationen zu Migrationsstrategien und zur Erstellung von Indizes finden Sie Migration zu Amazon DocumentDB in der Amazon DocumentDB-Dokumentation und im Blog: Migrieren Sie von MongoDB zu Amazon DocumentDB using the offline method.

Wenn beim Erstellen von Indizes in einem vorhandenen Cluster die Indexerstellung länger dauert als erwartet oder fehlschlägt, empfehlen wir, dass Sie die Instanz hochskalieren, um den Index zu erstellen, und dann, nachdem der Index erstellt wurde, wieder nach unten skalieren. Amazon DocumentDB ermöglicht es Ihnen, Instance-Größen innerhalb von Minuten schnell zu skalieren, indem Sie die AWS Management Console oder die AWS CLI verwenden. Weitere Informationen finden Sie unter Verwalten von Instance-Klassen. Bei der sekundengenauen Instance-Preisgestaltung zahlen Sie auf die Sekunde genau nur für die Ressourcen, die Sie verwenden.

Latenzprobleme bei der Erstellung des Hintergrundindexes und Fehler

Index-Builds im Hintergrund in Amazon DocumentDB werden erst gestartet, wenn alle Abfragen auf der primären Instance, die vor der Initiierung der Indexerstellung gestartet wurden, vollständig ausgeführt wurden. Wenn eine Abfrage mit langer Laufzeit läuft, werden Indexgenerierungen im Hintergrund blockiert, bis die Abfrage abgeschlossen ist, und es kann daher länger dauern als erwartet, bis sie abgeschlossen ist. Dies gilt auch dann, wenn Sammlungen leer sind.

Vordergrundindex-Builds weisen nicht dasselbe Blockierungsverhalten auf. Stattdessen sperren Index-Builds im Vordergrund die Sammlung exklusiv, bis die Indexerstellung abgeschlossen ist. Um Indizes für eine leere Sammlung zu erstellen und um das Blockieren lang andauernder Abfragen zu vermeiden, empfehlen wir daher, Indexbuilds im Vordergrund zu verwenden.

> **Note**
>
> Amazon DocumentDB erlaubt zu einem bestimmten Zeitpunkt nur die Erstellung eines Hintergrundindexes für eine Sammlung. Wenn DDL (Data Definition Language) Operationen

wie `createIndex()` oder `dropIndex()` während eines Indexaufbaus im HIntergrund in derselben Sammlung auftreten, schlägt der Aufbau des Hintergrundindex fehl.

Leistung und Ressourcenauslastung

Dieser Abschnitt enthält Fragen und Lösungen für häufig auftretende Diagnoseprobleme in Amazon DocumentDB-Bereitstellungen. Die bereitgestellten Beispiele verwenden die mongo-Shell und gelten für eine einzelne Instance. Informationen zum Suchen nach einem Instance-Endpunkt finden Sie unter Grundlegendes zu Amazon DocumentDB-Endpunkten.

Themen

Wie ermittle ich die Anzahl der Einfüge-, Aktualisierungs- und Löschvorgänge, die für meine Sammlung über die Mongo-API ausgeführt werden?

Um die Anzahl der Einfüge-, Aktualisierungs- und Löschvorgänge anzuzeigen, die für eine bestimmte Sammlung ausgeführt werden, führen Sie den folgenden Befehl für diese Sammlung aus:

```
db.collection.stats()
```

Die Ausgabe dieses Befehls beschreibt Folgendes unter seinem `opCounters` Feld:

- numDocsIns – Die Anzahl der Dokumente, die in diese Sammlung eingefügt wurden. Dazu gehören Dokumente, die mit den `insertMany` Befehlen `insert` und eingefügt wurden, sowie Dokumente, die durch ein Upsert eingefügt wurden.
- numDocsUpd – Die Anzahl der Dokumentenaktualisierungen in dieser Sammlung. Dazu gehören Dokumente, die mit den `findAndModify` Befehlen `update` und aktualisiert wurden.
- numDocsDel – Die Anzahl der Dokumente, die aus dieser Sammlung gelöscht wurden. Dazu gehören Dokumente`deleteMany`, die mit den `findAndModify` Befehlen `deleteOne`, `remove`, und gelöscht wurden.
- lastReset – Die Zeit, zu der diese Zähler zuletzt zurückgesetzt wurden. Die von diesem Befehl bereitgestellten Statistiken werden zurückgesetzt, wenn der Cluster gestartet/gestoppt oder die Instance hoch-/herunterskaliert wird.

Eine Beispielausgabe von , die ausgeführt wird, `db.collection.stats()` wird unten gezeigt.

```
{
    "ns" : "db.test",
    "count" : ...,
    "size" : ...,
    "avgObjSize" : ...,
    "storageSize" : ...,
    "capped" : false,
    "nindexes" : ...,
    "totalIndexSize" : ...,
    "indexSizes" : {
        "_id_" : ...,
        "x_1" : ...
```

```
    },
    "collScans" : ...,
    "idxScans" : ...,
    "opCounter" : {
        "numDocsIns" : ...,
        "numDocsUpd" : ...,
        "numDocsDel" : ...
    },
    "cacheStats" : {
        "collBlksHit" : ...,
        "collBlksRead" : ..,
        "collHitRatio" : ...,
        "idxBlksHit" : ...,
        "idxBlksRead" : ...,
        "idxHitRatio" : ...
    },
    "lastReset" : "2022-09-02 19:41:40.471473+00",
    "ok" : 1,
    "operationTime" : Timestamp(1662159707, 1)
}
```

Dieser Statistikbefehl sollte verwendet werden, wenn sammlungsspezifische Zähler für Einfüge-, Aktualisierungs- und Löschvorgänge über die Mongo-API angezeigt werden. Eine andere Möglichkeit, sammlungsspezifische Operationszähler anzuzeigen, besteht darin, die DML-Prüfung zu aktivieren. Die Anzahl der Einfüge-, Aktualisierungs- und Löschoperationen für alle Sammlungen in Zeitintervallen von einer Minute kann in angezeigt werden[Überwachen von Amazon DocumentDB mit CloudWatch](#).

Wie analysiere ich die Cache-Leistung?

Die Analyse der Cache-Leistung kann Einblicke in die Effizienz des Datenabrufs und der Systemleistung geben und basiert darauf, wie viele Daten von der Festplatte im Vergleich zum Cache gelesen werden. Wir stellen Cache-Statistiken über die Anzahl der Cache-Treffer (Daten, die aus dem Cache gelesen werden) und Cache-Fehler (Daten, die nicht im Cache gefunden und von der Festplatte gelesen werden) bereit, um Einblicke in die Cache-Leistung zu erhalten. Die Cache-Statistiken für eine bestimmte Sammlung finden Sie, indem Sie den folgenden Befehl für diese Sammlung ausführen:

```
db.collection.stats()
```

Die Werte im `cacheStats` Feld in der Ausgabe dieses Befehls stellen Cache-Statistiken für die Sammlung sowie die gesamten Cache-Statistiken für die Indizes bereit, die für die Sammlung erstellt wurden. Diese Statistiken sind unten aufgeführt:

- **`collBlksHit`** – Die Anzahl der Blöcke, die während Operationen in dieser Sammlung aus dem Cache gelesen wurden.
- **`collBlksRead`** – Die Anzahl der Blöcke, die während Operationen in dieser Sammlung von der Festplatte gelesen wurden (Cache-Fehler).
- **`collHitRatio`** – Die Cache-Trefferrate für diese Sammlung (`100 * [collBlksHit / (collBlksHit + collBlksRead)]`).
- **`idxBlksHit`** – Die Anzahl der Blöcke, die für jeden Index, der in dieser Sammlung erstellt wurde, aus dem Cache gelesen wurden.
- **`idxBlksRead`** – Die Anzahl der von der Festplatte gelesenen Blöcke (Cache-Fehler) für jeden Index, der in dieser Sammlung erstellt wurde.
- **`idxHitRatio`** – Die Cache-Trefferquote für die Indizes, die in dieser Sammlung erstellt wurden (`100 * [idxBlksHit / (idxBlksHit + idxBlksRead)]`).
- **`lastReset`** – Die Zeit, zu der diese Statistiken zuletzt zurückgesetzt wurden. Die von bereitgestellten Statistiken `db.collection.stats()` werden zurückgesetzt, wenn der Cluster gestartet/gestoppt oder die Instance hoch-/herunterskaliert wird.

Eine Aufschlüsselung der `idxBlksRead` Felder `idxBlksHit` und für jeden Index finden Sie auch mit dem `indexStats` Befehl . Indexspezifische Cache-Statistiken finden Sie, indem Sie den folgenden Befehl ausführen:

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Für jeden Index finden Sie die folgenden Cache-Statistiken unter dem `cacheStats` Feld :

- **`blksHit`** – Die Anzahl der Blöcke, die für diesen Index aus dem Cache gelesen wurden.
- **`blksRead`** – Die Anzahl der Blöcke, die für diesen Index von der Festplatte gelesen wurden.
- **`blksHitRatio`** – Die Cache-Trefferrate auf vier Dezimalstellen gerundet, berechnet durch `100 * [blksHit / (blksHit + blksRead)]`.

Wie finde und beende ich langsame und blockierte Abfragen?

Benutzerabfragen können aufgrund eines nicht optimalen Abfrageplans langsam ausgeführt oder aufgrund von Ressourcenkonflikten blockiert werden.

Zum Suchen nach Abfragen, die aufgrund eines nicht optimalen Abfrageplans verlangsamt oder aufgrund von Ressourcenkonflikten blockiert werden, verwenden Sie den Befehl `currentOp`. Sie können den Befehl filtern, um die Liste der relevanten Abfragen einzugrenzen. Mit der langsamen Abfrage muss `opid` verknüpft sein, um sie beenden zu können.

Die folgende Abfrage verwendet den Befehl `currentOp`, um alle Abfragen aufzulisten, die blockiert oder länger als 10 Sekunden ausgeführt werden.

```
db.adminCommand({
    aggregate: 1,
    pipeline: [
        {$currentOp: {}},
        {$match:
            {$or: [
                {secs_running: {$gt: 10}},
                {WaitState: {$exists: true}}]}},
        {$project: {_id:0, opid: 1, secs_running: 1}}],
    cursor: {}
});
```

Als Nächstes können Sie die Abfrage einschränken, um die `opid` von länger als 10 Sekunden ausgeführten Abfragen zu finden und diese zu beenden.

So finden und beenden Sie eine Abfrage, die länger als 10 Sekunden läuft:

1. Suchen Sie die `opid` der Abfrage.

   ```
   db.adminCommand({
       aggregate: 1,
       pipeline: [
           {$currentOp: {}},
           {$match:
               {$or:
                   [{secs_running: {$gt: 10}},
                    {WaitState: {$exists: true}}]}}],
       cursor: {}
   ```

```
});
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "opid" : 24646,
                "secs_running" : 12
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

2. Beenden Sie die Abfrage mit der Operation `killOp`.

```
db.adminCommand({killOp: 1, op: 24646});
```

Wie kann ich einen Abfrageplan sehen und eine Abfrage optimieren?

Wenn eine Abfrage langsam ausgeführt wird, erfordert die Abfrageausführung möglicherweise den vollständigen Scan der Sammlung, um die relevanten Dokumente auszuwählen. Manchmal kann die Abfrage schneller ausgeführt werden, wenn geeignete Indizes erstellt werden. Mit dem Befehl `explain` erkennen Sie dieses Szenario und können die Felder für die Indizes auswählen.

> **Note**
>
> Amazon DocumentDB emuliert die MongoDB 3.6-API auf einer speziell entwickelten Datenbank-Engine, die ein verteiltes, fehlertolerantes, selbstverstärkendes Speichersystem verwendet. Daher `explain()` können Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterschiedlich sein. Kunden, die die Kontrolle über ihren Abfrageplan wünschen, können den `$hint`-Operator verwenden, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Führen Sie die Abfrage, die Sie verbessern möchten, unter dem Befehl `explain` wie folgt aus.

```
db.runCommand({explain: {<query document>}})
```

Im Folgenden finden Sie eine Beispieloperation.

```
db.runCommand({explain:{
    aggregate: "sample-document",
    pipeline: [{$match: {x: {$eq: 1}}}],
    cursor: {batchSize: 1}}
});
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "db.test",
        "winningPlan" : {
            "stage" : "COLLSCAN"
        }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
}
```

Die Ausgabe oben zeigt an, dass in der Phase `$match` die gesamte Sammlung gescannt werden muss, um zu prüfen, ob das Feld `"x"` in jedem Dokument gleich 1 ist. Wenn die Sammlung zahlreiche Dokumente enthält, wird der Scan der Sammlung sehr langsam ausgeführt. Damit ist auch die Gesamtleistung der Abfrage sehr niedrig. Das Vorhandensein von `"COLLSCAN"` in der Ausgabe des Befehls `explain` zeigt daher an, dass die Abfrageleistung durch die Erstellung geeigneter Indizes verbessert werden kann.

In diesem Beispiel prüft die Abfrage, ob das Feld `"x"` in allen Dokumenten gleich 1 ist. Das Erstellen eines Indexes für das Feld `"x"` ermöglicht der Abfrage, einen vollständigen Scan der Sammlung zu vermeiden und den Index zu verwenden, um die relevanten Dokumente schneller zurückzugeben.

Nach dem Erstellen eines Indexes für das Feld `"x"` sieht die `explain`-Ausgabe wie folgt aus.

```
{
    "queryPlanner" : {
         "plannerVersion" : 1,
         "namespace" : "db.test",
         "winningPlan" : {
             "stage" : "IXSCAN",
             "indexName" : "x_1",
             "direction" : "forward"
         }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
}
```

Das Erstellen eines Indexes für das Feld `"x"` ermöglicht der Stufe `$match` die Verwendung eines Index, um die Anzahl der Dokumente zu reduzieren, in denen das Prädikat `"x = 1"` ausgewertet werden muss.

Bei kleinen Sammlungen kann der Amazon DocumentDB-Abfrageprozessor wählen, keinen Index zu verwenden, wenn die Leistungssteigerungen vernachlässigbar sind.

Wie kann ich einen Abfrageplan in elastischen Clustern sehen?

Um einen Abfrageplan in elastischen Clustern zu untersuchen, verwenden Sie den `explain` Befehl . Im Folgenden finden Sie eine `explain` Beispieloperation für eine Suchabfrage, die auf eine Sharded-Sammlung abzielt:

```
db.runCommand(
   {
     explain: { find: "cities", filter: {"name": "Seoul"}}
   }
)
```

> **Note**
>
> Amazon DocumentDB emuliert MongoDB auf einer speziell entwickelten Datenbank-Engine. Daher `explain()` können Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterschiedlich sein. Sie können den Abfrageplan mithilfe des -`$hint`Operators steuern, um die Auswahl eines bevorzugten Index zu erzwingen.

Die Ausgabe dieser Operation kann etwa wie folgt aussehen (JSON-Format):

```
{
  "queryPlanner" : {
    "elasticPlannerVersion" : 1,
    "winningPlan" : {
      "stage" : "SINGLE_SHARD",
      "shards" : [
        {
          "plannerVersion" : 1,
          "namespace" : "population.cities",
          "winningPlan" : {
            "stage" : "SHARD_MERGE",
            "shards" : [
              {
                "shardName" : "f2cf5cfd-fe9c-40ca-b4e5-298ca0d11111",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
                  "inputStages" : [
                    {
                      "stage" : "COLLSCAN",
                      "partitionCount" : 21
                    }
                  ]
                }
              },
              {
                "shardName" : "8f3f80e2-f96c-446e-8e9d-aab8c7f22222",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
```

```
                "inputStages" : [
                  {
                    "stage" : "COLLSCAN",
                    "partitionCount" : 21
                  }
                ]
              }
            },
            {
              "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a033333",
              "plannerVersion" : 1,
              "namespace" : "population.cities",
              "winningPlan" : {
                "stage" : "PARTITION_MERGE",
                "inputStages" : [
                  {
                    "stage" : "COLLSCAN",
                    "partitionCount" : 22
                  }
                ]
              }
            }
          ]
        },
        "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a0f3fb58"
      }
    ]
  }
},
"serverInfo" : {
  "host" : "example-4788267630.us-east-1.docdb-elastic.amazonaws.com:27017",
  "version" : "5.0.0"
},
"ok" : 1,
"operationTime" : Timestamp(1695097923, 1)
}
```

Die obige Ausgabe zeigt den Abfrageplan für die `find` Abfrage auf einem Cluster mit drei Shards. Jeder Shard hat mehrere Datenpartitionen, die unterschiedliche Eingabephasen haben können. In diesem Beispiel wird ein „COLLSCAN“ (ein Sammlungsscan) auf allen Partitionen ausgeführt, bevor die Ergebnisse in der Phase „PARTITION_MERGE“ innerhalb jedes

Shards zusammengeführt werden. Die Ergebnisse für alle Shards werden dann in der Phase „SHARD_MERGE“ zusammengeführt, bevor sie zurück an den Client gesendet werden.

Wie liste ich alle laufenden Operationen für eine Instance auf?

Als Benutzer oder Primärbenutzer möchten Sie häufig alle aktuellen Operationen auflisten, die auf einer Instance ausgeführt werden, um Diagnose- und Fehlerbehebungszwecke zu verwenden. (Weitere Informationen zum Verwalten von Benutzern finden Sie unter Amazon-DocumentDB-Benutzer.)

Mit der mongo Shell können Sie die folgende Abfrage verwenden, um alle laufenden Operationen auf einer Amazon DocumentDB-Instance aufzulisten.

```
db.adminCommand({currentOp: 1, $all: 1});
```

Die Abfrage gibt die vollständige Liste aller Benutzerabfragen und internen Systemaufgaben zurück, die zurzeit auf der Instance ausgeführt werden.

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "inprog" : [
        {
            "desc" : "INTERNAL"
        },
        {
            "desc" : "TTLMonitor",
            "active" : false
        },
        {
            "client" : ...,
            "desc" : "Conn",
            "active" : true,
            "killPending" : false,
            "opid" : 195,
            "ns" : "admin.$cmd",
            "command" : {
                "currentOp" : 1,
                "$all" : 1
            },
            "op" : "command",
```

```
            "$db" : "admin",
            "secs_running" : 0,
            "microsecs_running" : NumberLong(68),
            "clientMetaData" : {
            "application" : {
                "name" : "MongoDB Shell"
            },
            "driver" : {
                ...
            },
            "os" : {
                ...
            }
         }
      },
      {
         "desc": "GARBAGE_COLLECTION",
         "garbageCollection": {
            "databaseName": "testdb",
            "collectionName": "testCollectionA"
         },
         "secs_running": 3,
         "microsecs_running": NumberLong(3123456)
      },
      {
         "desc": "GARBAGE_COLLECTION",
         "garbageCollection": {
            "databaseName": "testdb",
            "collectionName": "testCollectionB"
         },
         "secs_running": 4,
         "microsecs_running": NumberLong(4123456)
      }
   ],
   "ok" : 1
}
```

Die folgenden Werte sind gültige Werte für das Feld `"desc"`:

- **`INTERNAL`** – Interne Systemaufgaben wie die Cursor-Bereinigungs- oder veraltete Benutzerbereinigungsaufgaben.
- **`TTLMonitor`** – Der TTL-Monitor-Thread (Time to Live). Der Ausführungsstatus wird im Feld `"active"` dargestellt.

- **`GARBAGE_COLLECTION`** – Der interne Garbage Collector-Thread.
- **`CONN`** – Die Benutzerabfrage.
- **`CURSOR`** – Die Operation ist ein inaktiver Cursor, der darauf wartet, dass der Benutzer den Befehl „getMore“ aufruft, um den nächsten Ergebnisstapel zu erhalten. In diesem Zustand verbraucht der Cursor Speicher, verbraucht jedoch keine Rechenleistung.

In der obigen Ausgabe werden auch alle Benutzerabfragen im System aufgeführt. Jede Benutzerabfrage wird im Kontext einer Datenbank und einer Sammlung ausgeführt. Die Kombination dieser beiden Komponenten wird als Namespace bezeichnet. Der Namespace jeder Benutzerabfrage ist im Feld `"ns"` verfügbar.

Manchmal müssen Sie alle Benutzerabfragen auflisten, die in einem bestimmten Namespace ausgeführt werden. Daher muss die vorherige Ausgabe anhand des Felds `"ns"` gefiltert werden. Im Folgenden finden Sie eine Beispielabfrage für eine gefilterte Ausgabe. Die Abfrage listet alle Benutzerabfragen auf, die zurzeit in der Datenbank `"db"` und in der Sammlung `"test"` ausgeführt werden (d. h. im Namespace `"db.test"`).

```
db.adminCommand({aggregate: 1,
    pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
               {$match: {ns: {$eq: "db.test"}}}],
    cursor: {}
});
```

Als primärer Benutzer des Systems können Sie Abfragen aller Benutzer und auch aller internen Systemaufgaben sehen. Alle anderen Benutzer können nur ihre jeweiligen Abfragen anzeigen.

Wenn die Gesamtzahl der Abfragen und internen Systemaufgaben die Standard-Batch-Cursor-Größe überschreitet, generiert die `mongo`-Shell automatisch das Iterator-Objekt `'it'`, um den Rest der Ergebnisse anzuzeigen. Führen Sie den Befehl `'it'` so lange aus, bis alle Ergebnisse angezeigt wurden.

Woher weiß ich, wann eine Abfrage ausgeführt wird?

Benutzerabfragen können aufgrund eines nicht optimalen Abfrageplans langsam ausgeführt oder aufgrund von Ressourcenkonflikten blockiert werden. Das Debuggen solcher Abfragen ist ein mehrstufiger Prozess, in dem möglicherweise mehrmals dieselben Schritte ausgeführt werden müssen.

Im ersten Debugging-Schritt werden alle langsamen oder blockierten Abfragen aufgeführt. Die folgende Abfrage listet alle Benutzerabfragen auf, die länger als 10 Sekunden ausgeführt wurden oder auf Ressourcen warten.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {}},
                            {$match: {$or: [{secs_running: {$gt: 10}},
                                            {WaitState: {$exists: true}}]}},
                            {$project: {_id:0,
                                        opid: 1,
                                        secs_running: 1,
                                        WaitState: 1,
                                        blockedOn: 1,
                                        command: 1}}],
                 cursor: {}
                });
```

Wiederholen Sie die vorherige Abfrage regelmäßig, um zu ermitteln, ob sich die Liste der Abfragen ändert, und die langsamen oder blockierten Abfragen zu identifizieren.

Wenn das Ausgabedokument der betreffenden Abfrage das Feld `WaitState` enthält, zeigt dies an, dass die Abfrage aufgrund von Ressourcenkonflikten langsam ist oder blockiert wird. Die Ressourcenkonflikte könnten auf E/A-Vorgänge, interne Systemaufgaben oder die Abfragen anderer Benutzer zurückgehen.

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "opid" : 201,
                "command" : {
                    "aggregate" : ...
                },
                "secs_running" : 208,
                "WaitState" : "IO"
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
```

```
    },
    "ok" : 1
}
```

Die E/A-Schnittstelle kann einen Engpass darstellen, wenn zahlreiche Abfragen in verschiedenen Sammlungen auf derselben Instance gleichzeitig ausgeführt werden oder die Instance zu klein für den Datensatz ist, auf dem die Abfrage ausgeführt wird. Wenn es sich bei den Abfragen um schreibgeschützte Abfragen handelt, können Sie der eben beschriebenen Situation durch Aufteilung der Abfragen für die einzelnen Sammlungen auf separate Replikate abhelfen. Wenn in verschiedenen Sammlungen gleichzeitig Updates ausgeführt werden oder die Instance zu klein für den Datensatz ist, können Sie die Instance aufwärts skalieren.

Wenn der Ressourcenkonflikt auf Abfragen anderer Benutzer zurückzuführen ist, zeigt das Feld `"blockedOn"` im Ausgabedokument die `"opid"` der Abfrage an, die sich auf die Abfrage auswirkt. Folgen Sie mithilfe der `"opid"` der Kette der Felder `"WaitState"` und `"blockedOn"` aller Abfragen, um die Abfrage an der Ende der Kette zu finden.

Wenn es sich bei der Aufgabe am Ende der Kette um eine interne Aufgabe handelt, kann das Problem nur behoben werden, indem die Abfrage beendet und später erneut ausgeführt wird.

Im Folgenden finden Sie eine Beispielausgabe, in der die Suchabfrage auf einer Sammlungssperre blockiert wird, die im Besitz einer anderen Aufgabe ist.

```
{
    "inprog" : [
        {
            "client" : "...",
            "desc" : "Conn",
            "active" : true,
            "killPending" : false,
            "opid" : 75,
            "ns" : "...",
            "command" : {
                "find" : "...",
                "filter" : {

                }
            },
            "op" : "query",
            "$db" : "test",
            "secs_running" : 9,
```

```
            "microsecs_running" : NumberLong(9449440),
            "threadId" : 24773,
            "clientMetaData" : {
                "application" : {
                   "name" : "MongoDB Shell"
                },
                "driver" : {
                    ...
                },
                "os" : {
                    ...
                }
            },
            "WaitState" : "CollectionLock",
            "blockedOn" : "INTERNAL"
        },
        {
            "desc" : "INTERNAL"
        },
        {
            "client" : "...",
            ...
            "command" : {
                "currentOp" : 1
            },
            ...
        }
    ],
    "ok" : 1
}
```

Wenn "WaitState" die Werte "Latch", "SystemLock", "BufferLock", "BackgroundActivity" oder "Other" hat, wird der Ressourcenkonflikt von internen Systemaufgaben verursacht. Wenn die Situation lange andauert, besteht die einzige Abhilfe darin, die Abfrage zu beenden und später erneut auszuführen.

Wie stelle ich fest, warum ein System plötzlich langsam ausgeführt wird?

Im Folgenden finden Sie einige häufige Gründe für die Verlangsamung von Systemen:

- Übermäßige Ressourcenkonflikte zwischen gleichzeitigen Abfragen
- Zunahme der Anzahl der aktiven gleichzeitigen Abfragen im Laufe der Zeit

- Interne Systemaufgaben wie "GARBAGE_COLLECTION"

Wenn Sie die Systemnutzung im Laufe der Zeit überwachen möchten, führen Sie in regelmäßigen Abständen die folgende "currentOp"-Abfrage aus, und geben Sie die Ergebnisse in einem externen Speicher aus. Die Abfrage zählt die Anzahl der Abfragen und Operationen in jedem Namespace im System. Sie können anschließend die Systemnutzungsergebnisse analysieren, um die Systemlast zu verstehen und entsprechende Maßnahmen zu ergreifen.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Diese Abfrage gibt eine Aggregation aller in den einzelnen Namespaces ausgeführten Abfragen, alle internen Systemaufgaben und die eindeutige Anzahl der Wartezustände (wenn vorhanden) pro Namespace zurück.

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "_id" : {
                    "desc" : "Conn",
                    "ns" : "db.test",
                    "WaitState" : "CollectionLock"
                },
               "count" : 2
            },
            {
                "_id" : {
                    "desc" : "Conn",
                    "ns" : "admin.$cmd"
                },
                "count" : 1
            },
            {
```

```
                "_id" : {
                    "desc" : "TTLMonitor"
                },
                "count" : 1
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

In der oben gezeigten Ausgabe werden zwei Benutzerabfragen im Namespace `"db.test"` durch eine Sammlungssperre blockiert: 1 Abfrage in Namespace `"admin.$cmd"` und eine interne `"TTLMonitor"`-Aufgabe.

Wenn die Ausgabe viele Abfragen mit blockierenden Wartezuständen anzeigt, finden Sie Informationen unter Wie finde und beende ich langsame und blockierte Abfragen?.

Wie ermittle ich die Ursache für eine hohe CPU-Auslastung auf einer oder mehreren Cluster-Instances?

In den folgenden Abschnitten finden Sie Informationen, die Ihnen möglicherweise helfen, die Ursache für die hohe CPU-Nutzung einer Instance zu ermitteln. Die Ergebnisse sind vom Workload abhängig.

- Informationen dazu, wie Sie ermitteln, warum eine Instance plötzlich langsam ausgeführt wird, finden Sie unter Wie stelle ich fest, warum ein System plötzlich langsam ausgeführt wird?.
- Informationen zum Identifizieren und Beenden langsamer Abfragen auf einer bestimmten Instance finden Sie unter Wie finde und beende ich langsame und blockierte Abfragen?.
- Informationen dazu, wie Sie ermitteln, ob eine Abfrage fortschreitet, finden Sie unter Woher weiß ich, wann eine Abfrage ausgeführt wird?.
- Informationen dazu, warum die Ausführung einer Abfrage lange dauert, finden Sie unter Wie kann ich einen Abfrageplan sehen und eine Abfrage optimieren?.
- Informationen dazu, wie Sie langsame Abfragen über die Zeit nachverfolgen, finden Sie unter Profilierung von Amazon DocumentDB-Vorgängen.

Abhängig vom Grund für die hohe CPU-Nutzung Ihrer Instance können Sie einen oder mehrere der folgenden Schritte ausführen.

- Wenn die primäre Instance eine hohe CPU-Nutzung aufweist, die Replikat-Instances jedoch nicht, sollten Sie den Datenverkehr für Lesevorgänge über die Client-Leseeinstellungen auf mehrere Replikate verteilen (z. B. `secondaryPreferred`). Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Amazon DocumentDB als Replikatsatz.

  Die Verwendung von Replikaten für Lesevorgänge kann zu einer besseren Nutzung der Cluster-Ressourcen führen, da die primäre Instance eine größere Menge an Schreibdatenverkehr verarbeiten kann. Lesevorgänge aus -Replikaten sind Eventually Consistent.
- Wenn die hohe CPU-Nutzung durch Schreib-Workloads verursacht wird, können Sie durch die Skalierung der Cluster-Instances auf einen größeren Instance-Typ die Anzahl der CPU-Kerne erhöhen, die für den Workload verfügbar sind. Weitere Informationen finden Sie unter Instances und Instance-Klassen-Spezifikationen.
- Wenn alle Cluster-Instances eine hohe CPU-Nutzung aufweisen und der Workload Replikate für Lesevorgänge verwendet, erhöht das Hinzufügen weiterer Replikate zum Cluster die Zahl der für Lesevorgänge verfügbaren Ressourcen. Weitere Informationen finden Sie unter Hinzufügen einer Amazon DocumentDB-Instance zu einem Cluster.

Wie ermittle ich die offenen Cursor auf einer Instance?

Wenn Sie mit einer Amazon DocumentDB-Instance verbunden sind, können Sie den Befehl verwenden, `db.runCommand("listCursors")` um die offenen Cursor auf dieser Instance aufzulisten. Je nach Instance-Typ sind bis zu 4 560 aktive Cursors auf einer bestimmten Amazon DocumentDB-Instance geöffnet. Es wird allgemein empfohlen, Cursor zu schließen, die nicht mehr verwendet werden, da Cursor Ressourcen auf einer Instance verwenden und eine Obergrenze haben. Spezifische Limits finden Sie unter Kontingente und Limits für Amazon DocumentDB .

```
db.runCommand("listCursors")
```

Wie ermittle ich die aktuelle Version der Amazon DocumentDB-Engine?

Führen Sie den folgenden Befehl aus, um Ihre aktuelle Amazon DocumentDB-Engine-Version zu ermitteln.

```
db.runCommand({getEngineVersion: 1})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format).

```
{ "engineVersion" : "2.x.x", "ok" : 1 }
```

> **Note**
>
> Die Engine-Version für Amazon DocumentDB 3.6 ist 1.x.x und die Engine-Version für Amazon DocumentDB 4.0 ist 2.x.x.

Wie analysiere ich die Indexnutzung und identifiziere ungenutzte Indizes?

Zum Identifizieren der Indizes für eine bestimmte Sammlung führen Sie den folgenden Befehl aus:

```
db.collection.getIndexes()
```

Um zu analysieren, wie viele Indizes bei Operationen verwendet werden, die an den Sammlungen ausgeführt werden, können die `indexStats` Befehle `collStats` und verwendet werden. Führen Sie den folgenden Befehl aus, um die Gesamtzahl der Scans anzuzeigen, die mit Indizes (Indexscans) im Vergleich zur Anzahl der Scans ohne Index (Sammlungsscans) durchgeführt wurden:

```
db.collection.stats()
```

Die Ausgabe für diesen Befehl enthält die folgenden Werte:

- **`idxScans`** – Die Anzahl der Scans, die mit einem Index für diese Sammlung durchgeführt wurden.
- **`collScans`** – Die Anzahl der Scans, die für diese Sammlung ohne Verwendung eines Index durchgeführt wurden. Diese Scans hätten dazu geführt, dass die Dokumente in der Sammlung nacheinander betrachtet wurden.
- **`lastReset`** – Die Zeit, zu der diese Zähler zuletzt zurückgesetzt wurden. Die von diesem Befehl bereitgestellten Statistiken werden zurückgesetzt, wenn der Cluster gestartet/gestoppt oder die Instance hoch-/herunterskaliert wird.

Eine Aufschlüsselung der verwendeten Indexe finden Sie in der Ausgabe des folgenden Befehls. Es hat sich bewährt, regelmäßig ungenutzte Indizes zu identifizieren und zu entfernen, um die Leistung zu verbessern und die Kosten zu senken, da unnötige Rechen-, Speicher- und E/A-Operationen vermieden werden, die zur Wartung der Indizes verwendet werden.

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Die Ausgabe dieses Befehls liefert die folgenden Werte für jeden Index, der in der Sammlung erstellt wurde:

- **`ops`** – Die Anzahl der Operationen, die den Index verwendet haben. Wenn Ihre Workload über einen ausreichend langen Zeitraum ausgeführt wurde und Sie sicher sind, dass sich Ihre Workload in einem konstanten Zustand befindet, würde ein `ops`-Wert von Null anzeigen, dass der Index überhaupt nicht verwendet wird.
- **`numDocsRead`** – Die Anzahl der Dokumente, die während Operationen mit diesem Index gelesen wurden.
- **`since`** – Die Zeit, seit Amazon DocumentDB mit der Erfassung von Statistiken zur Indexnutzung begonnen hat, was in der Regel der Wert seit dem letzten Neustart oder der letzten Wartungsaktion der Datenbank ist.
- **`size`** – Die Größe dieses Index in Byte.

Das folgende Beispiel ist eine Beispielausgabe der Ausführung des obigen Befehls:

```
{
    "name" : "_id_",
    "key" : {
        "_id" : 1
    },
    "host" : "example-host.com:12345",
    "size" : NumberLong(...),
    "accesses" : {
        "ops" : NumberLong(...),
        "docsRead" : NumberLong(...),
        "since" : ISODate("...")
    },
    "cacheStats" : {
        "blksRead" : NumberLong(...),
        "blksHit" : NumberLong(...),
        "hitRatio" : ...
    }
}
{
    "name" : "x_1",
    "key" : {
```

```
        "x" : 1
    },
    "host" : "example-host.com:12345",
    "size" : NumberLong(...),
    "accesses" : {
        "ops" : NumberLong(...),
        "docsRead" : NumberLong(...),
        "since" : ISODate("...")
    },
    "cacheStats" : {
        "blksRead" : NumberLong(...),
        "blksHit" : NumberLong(...),
        "hitRatio" : ...
    }
}
```

Um die Gesamtindexgröße für eine Sammlung zu ermitteln, führen Sie den folgenden Befehl aus:

```
db.collection.stats()
```

Führen Sie den folgenden Befehl aus, um einen nicht verwendeten Index zu löschen:

```
db.collection.dropIndex("indexName")
```

Wie erkenne ich fehlende Indizes?

Sie können den Amazon DocumentDB-Profiler verwenden, um langsame Abfragen zu protokollieren. Eine Abfrage, die wiederholt im Protokoll über langsame Abfragen erscheint, kann darauf hinweisen, dass ein zusätzlicher Index erforderlich ist, um diese Abfrageleistung zu verbessern.

Sie können Möglichkeiten für hilfreiche Indizes identifizieren, indem Sie nach lang laufenden Abfragen suchen, die eine oder mehrere Stufen haben und mindestens eine `COLLSCAN`-Stufe durchführen, d.h. in der Abfragephase muss jedes Dokument in der Sammlung gelesen werden, um eine Antwort auf die Abfrage bereitzustellen.

Das folgende Beispiel zeigt eine Abfrage zu einer Sammlung von Taxifahrten, die auf einer großen Sammlung ausgeführt wurden.

```
db.rides.count({"fare.totalAmount":{$gt:10.0}}))
```

Um dieses Beispiel auszuführen, musste die Abfrage einen Sammlungsscan durchführen (d. h. jedes einzelne Dokument in der Sammlung lesen), da es keinen Index für das `fare.totalAmount`-Feld gibt. Die Ausgabe vom Amazon DocumentDB-Profiler für diese Abfrage sieht etwa wie folgt aus:

```
{
    ...
    "cursorExhausted": true,
    "nreturned": 0,
    "responseLength": 0,
    "protocol": "op_query",
    "millis": 300679,
    "planSummary": "COLLSCAN",
    "execStats": {
        "stage": "COLLSCAN",
        "nReturned": "0",
        "executionTimeMillisEstimate": "300678.042"
    },
    "client": "172.31.5.63:53878",
    "appName": "MongoDB Shell",
    "user": "example"
}
```

Um die Abfrage in diesem Beispiel zu beschleunigen, möchten Sie einen Index auf `fare.totalAmount` erstellen, wie unten gezeigt.

```
db.rides.createIndex( {"fare.totalAmount": 1}, {background: true} )
```

> **Note**
>
> Indizes, die im Vordergrund erstellt werden (d. h. wenn die Option `{background:true}` beim Erstellen des Index nicht angegeben wurde), erhalten eine exklusive Schreibsperre, die Anwendungen daran hindert, Daten so lange in die Sammlung zu schreiben, bis der Indexaufbau abgeschlossen ist. Beachten Sie diese mögliche Auswirkung beim Erstellen von Indizes für Produktions-Cluster. Beim Erstellen von Indizes empfehlen wir die Einstellung `{background:true}`.

Im Allgemeinen möchten Sie Indizes für Felder mit hoher Kardinalität erstellen (z. B. eine große Anzahl eindeutiger Werte). Das Erstellen eines Index für ein Feld mit geringer Kardinalität kann zu einem großen Index führen, der nicht verwendet wird. Der Amazon DocumentDB-Abfrageoptimierer

berücksichtigt die Gesamtgröße der Sammlung und die Selektivität der Indizes beim Erstellen eines Abfrageplans. Manchmal wird der Abfrageprozessor einen `COLLSCAN` auswählen, selbst wenn ein Index vorhanden ist. Dies geschieht, wenn der Abfrageprozessor schätzt, dass die Verwendung des Index keinen Leistungsvorteil gegenüber dem Scannen der gesamten Sammlung bringt. Wenn Sie den Abfrageprozessor zwingen möchten, einen bestimmten Index zu verwenden, können Sie den `hint()`-Operator wie unten gezeigt verwenden.

```
db.collection.find().hint("indexName")
```

Zusammenfassung nützlicher Abfragen

Die folgenden Abfragen können für die Überwachung der Leistung und Ressourcenauslastung in Amazon DocumentDB nützlich sein.

- Verwenden Sie den folgenden Befehl, um Statistiken zu einer bestimmten Sammlung anzuzeigen, einschließlich Operationszähler, Cache-Statistiken, Zugriffsstatistiken und Größenstatistiken:

  ```
  db.collection.stats()
  ```

- Verwenden Sie den folgenden Befehl, um Statistiken zu jedem Index anzuzeigen, der für eine Sammlung erstellt wurde, einschließlich der Größe des Index, indexspezifischer Cache-Statistiken und Indexnutzungsstatistiken:

  ```
  db.collection.aggregate([{$indexStats:{}}]).pretty()
  ```

- Verwenden Sie die folgende Abfrage, um alle Aktivitäten aufzulisten.

  ```
  db.adminCommand({currentOp: 1, $all: 1});
  ```

- Der folgende Code listet alle langsamen oder blockierten Abfragen auf.

  ```
  db.adminCommand({aggregate: 1,
                   pipeline: [{$currentOp: {}},
                              {$match: {$or: [{secs_running: {$gt: 10}},
                                              {WaitState: {$exists: true}}]}},
                              {$project: {_id:0,
                                          opid: 1,
                                          secs_running: 1,
                                          WaitState: 1,
                                          blockedOn: 1,
  ```

```
                                        command: 1}}],
                 cursor: {}
                });
```

- Der folgende Code beendet eine Abfrage.

```
db.adminCommand({killOp: 1, op: <opid of running or blocked query>});
```

- Verwenden Sie den folgenden Code, um eine aggregierte Ansicht des Systemstatus zu erhalten.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

API-Referenz für die Amazon DocumentDB Document-DB-Cluster-, Instance- und Ressourc

Dieser Abschnitt beschreibt die Cluster-, Instance- und Ressourcenverwaltungsoperationen für Amazon DocumentDB (mit Mongo-DB-Kompatibilität), auf die über HTTP zugegriffen werden kannAWS Command Line Interface(AWS CLI), oder dasAWS-SDK-SDKS. Sie können diese APIs zum Erstellen, Löschen und Ändern von Clustern und Instances verwenden.

> **Important**
>
> Diese APIs werden nur für die Verwaltung von Clustern, Instances und verwandten Ressourcen verwendet. Weitere Informationen zum Herstellen einer Verbindung mit einem laufenden Amazon Document-DB-Cluster finden Sie unterHandbuch „Erste Schritte“aus.

Themen

Aktionen

Die folgenden Aktionen werden von unterstützt Amazon DocumentDB (with MongoDB compatibility):

- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster
- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters
- DescribeDBClusterSnapshotAttributes
- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- ListTagsForResource
- ModifyDBCluster

- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster

Die folgenden Aktionen werden von Amazon DocumentDB Elastic Clustern unterstützt:

- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster
- GetClusterSnapshot
- ListClusters
- ListClusterSnapshots
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster

- TagResource
- UntagResource
- UpdateCluster

Amazon DocumentDB (with MongoDB compatibility)

Folgende Aktionen werden unterstützt vonAmazon DocumentDB (with MongoDB compatibility):

- AddSourceIdentifierToSubscription
- AddTagsToResource
- ApplyPendingMaintenanceAction
- CopyDBClusterParameterGroup
- CopyDBClusterSnapshot
- CreateDBCluster
- CreateDBClusterParameterGroup
- CreateDBClusterSnapshot
- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster
- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters
- DescribeDBClusterSnapshotAttributes

- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- ListTagsForResource
- ModifyDBCluster
- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster

AddSourceIdentifierToSubscription

Service: Amazon DocumentDB (with MongoDB compatibility)

Fügt eine Quell-ID einem Abonnement für Ereignisbenachrichtigungen hinzu.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceIdentifier

Die Kennung der hinzuzufügenden Ereignisquelle:

- Wenn der Quelltyp eine Instance ist, `DBInstanceIdentifier` muss ein angegeben werden.
- Wenn der Quelltyp eine Sicherheitsgruppe ist, `DBSecurityGroupName` muss ein angegeben werden.
- Wenn der Quelltyp eine Parametergruppe ist, `DBParameterGroupName` muss ein angegeben werden.
- Wenn der Quelltyp ein Snapshot ist, `DBSnapshotIdentifier` muss ein angegeben werden.

Typ: Zeichenfolge

Erforderlich: Ja

SubscriptionName

Der Name des Abonnements für Amazon DocumentDB-Ereignisbenachrichtigungen, dem Sie eine Quell-ID hinzufügen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einem Ereignis, das Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

SourceNotFound

Die angeforderte Quelle konnte nicht gefunden werden.

HTTP Status Code: 404

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

AddTagsToResource

Service: Amazon DocumentDB (with MongoDB compatibility)

Fügt Metadaten-Tags zu einer Amazon DocumentDB-Ressource hinzu. Sie können diese Tags mit Kostenzuordnungsberichten verwenden, um Kosten zu verfolgen, die mit Amazon DocumentDB-Ressourcen oder in einer -`Condition`Anweisung in einer AWS Identity and Access Management (IAM)-Richtlinie für Amazon DocumentDB verbunden sind.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ResourceName

Die Amazon DocumentDB-Ressource, der die Tags hinzugefügt werden. Dieser Wert ist ein Amazon-Ressourcenname .

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die der Amazon DocumentDB-Ressource zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ApplyPendingMaintenanceAction

Service: Amazon DocumentDB (with MongoDB compatibility)

Wendet eine ausstehende Wartungsaktion auf eine Ressource an (z. B. auf eine Amazon DocumentDB-Instance).

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ApplyAction

Die anstehende Wartungsaktion, die auf diese Ressource angewendet werden soll.

Zulässige Werte: `system-update`, `db-upgrade`

Typ: Zeichenfolge

Erforderlich: Ja

OptInType

Ein Wert, der den Typ der Opt-In-Anforderung angibt oder eine Opt-In-Anforderung rückgängig macht. Eine Opt-in-Anfrage vom Typ `immediate` kann nicht rückgängig gemacht werden.

Zulässige Werte:

- `immediate` – Die Wartungsmaßnahme sofort anwenden.
- `next-maintenance` – Die Wartungsaktion im nächsten Wartungsfenster für die Ressource anwenden.
- `undo-opt-in` – Alle bestehenden `next-maintenance`-Opt-In-Anfragen stornieren.

Typ: Zeichenfolge

Erforderlich: Ja

ResourceIdentifier

Der Amazon-Ressourcenname (ARN) der Ressource, auf die sich die anstehende Wartungsaktion bezieht.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

ResourcePendingMaintenanceActions

Stellt die Ausgabe von darApplyPendingMaintenanceAction.

Typ: ResourcePendingMaintenanceActions Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

ResourceNotFoundFault

Die angegebene Ressourcen-ID wurde nicht gefunden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CopyDBClusterParameterGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Kopiert die angegebene Cluster-Parametergruppe.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceDBClusterParameterGroupIdentifier

Die Kennung oder der Amazon-Ressourcenname (ARN) für die Quell-Cluster-Parametergruppe.

Einschränkungen:

- Muss eine gültige Cluster-Parametergruppe angeben.
- Wenn sich die Quell-Cluster-Parametergruppe in derselben AWS-Region wie die Kopie befindet, geben Sie eine gültige Parametergruppen-ID an, z. B. `my-db-cluster-param-group`oder einen gültigen ARN.
- Wenn sich die Quellparametergruppe in einer anderen AWS-Region als die Kopie befindet, geben Sie einen gültigen Cluster-Parametergruppen-ARN an, z. B. `arn:aws:rds:us-east-1:123456789012:sample-cluster:sample-parameter-group`.

Typ: Zeichenfolge

Erforderlich: Ja

TargetDBClusterParameterGroupDescription

Eine Beschreibung für die kopierte Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

TargetDBClusterParameterGroupIdentifier

Die Kennung für die kopierte Cluster-Parametergruppe.

Einschränkungen:

- Kann nicht Null, leer oder negativ sein.
- Muss zwischen 1 und 255 Buchstaben, Ziffern oder Bindestriche enthalten.

- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster-param-group1`

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die der Parametergruppe zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterParameterGroup

Detaillierte Informationen zu einer Cluster-Parametergruppe.

Typ: DBClusterParameterGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupAlreadyExists

Eine Parametergruppe mit demselben Namen ist bereits vorhanden.

HTTP Status Code: 400

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

DBParameterGroupQuotaExceeded

Diese Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Parametergruppen überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CopyDBClusterSnapshot

Service: Amazon DocumentDB (with MongoDB compatibility)

Kopiert einen Snapshot eines Clusters.

Um einen Cluster-Snapshot aus einem freigegebenen manuellen Cluster-Snapshot zu kopieren, `SourceDBClusterSnapshotIdentifier` muss der Amazon-Ressourcenname (ARN) des freigegebenen Cluster-Snapshots sein. Sie können einen freigegebenen DB-Cluster-Snapshot, ob verschlüsselt oder nicht, nur in dieselbe AWS-Region kopieren.

Um den Kopiervorgang abzubrechen, nachdem er ausgeführt wurde, löschen Sie den von identifizierten Ziel-Cluster-Snapshot, `TargetDBClusterSnapshotIdentifier` während sich dieser Cluster-Snapshot im Kopierstatus befindet.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceDBClusterSnapshotIdentifier

Die ID des zu kopierenden Cluster-Snapshots. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Es ist ein gültiger System-Snapshot im Status verfügbar erforderlich.
- Wenn sich der Quell-Snapshot in derselben AWS-Region wie die Kopie befindet, geben Sie eine gültige Snapshot-ID an.
- Wenn sich der Quell-Snapshot in einer anderen AWS-Region als die Kopie befindet, geben Sie einen gültigen Cluster-Snapshot-ARN an.

Beispiel: `my-cluster-snapshot1`

Typ: Zeichenfolge

Erforderlich: Ja

TargetDBClusterSnapshotIdentifier

Die ID des neuen Cluster-Snapshots, der aus dem Quell-Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster-snapshot2`

Typ: Zeichenfolge

Erforderlich: Ja

CopyTags

Legen Sie den Wert auf fest, `true` um alle Tags aus dem Quell-Cluster-Snapshot in den Ziel-Cluster-Snapshot zu kopieren, andernfalls `false`. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüssel-ID für einen verschlüsselten Cluster-Snapshot. Die AWS KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN), die AWS KMS-Schlüssel-ID oder der AWS KMS-Schlüsselalias für den AWS KMS-Verschlüsselungsschlüssel.

Wenn Sie einen verschlüsselten Cluster-Snapshot aus Ihrem kopierenAWS-Konto, können Sie einen Wert für angeben, `KmsKeyId` um die Kopie mit einem neuen AWS KMSVerschlüsselungsschlüssel zu verschlüsseln. Wenn Sie keinen Wert für angeben`KmsKeyId`, wird die Kopie des Cluster-Snapshots mit demselben AWS KMS Schlüssel wie der Quell-Cluster-Snapshot verschlüsselt.

Wenn Sie einen verschlüsselten Cluster-Snapshot kopieren, der von einem anderen freigegeben wirdAWS-Konto, müssen Sie einen Wert für angeben`KmsKeyId`.

Um einen verschlüsselten Cluster-Snapshot in ein anderes zu kopierenAWS-Region, setzen Sie `KmsKeyId` auf die AWS KMS Schlüssel-ID, die Sie zum Verschlüsseln der Kopie des Cluster-Snapshots in der -Zielregion verwenden möchten. -AWS KMSVerschlüsselungsschlüssel sind spezifisch für die AWS-Region , in der sie erstellt werden, und Sie können keine Verschlüsselungsschlüssel von einem AWS-Region in einem anderen verwendenAWS-Region.

Wenn Sie einen unverschlüsselten Cluster-Snapshot kopieren und einen Wert für den `KmsKeyId` Parameter angeben, wird ein Fehler zurückgegeben.

Typ: Zeichenfolge

Erforderlich: Nein

PreSignedUrl

Die URL, die eine mit Signature Version 4 signierte Anforderung für die`CopyDBClusterSnapshot` API-Aktion in der enthältAWS-Region, die den zu kopierenden Quell-Cluster-Snapshot enthält. Sie müssen den `PreSignedUrl` Parameter verwenden, wenn Sie einen Cluster-Snapshot aus einem anderen kopierenAWS-Region.

Wenn Sie ein AWS SDK-Tool oder die verwendenAWS CLI, können Sie `SourceRegion` (oder `--source-region` für die AWS CLI) angeben, anstatt `PreSignedUrl` manuell anzugeben. Durch die Angabe von wird `SourceRegion` automatisch eine vorsignierte URL generiert, die eine gültige Anforderung für die Operation ist, die in der Quelle ausgeführt werden kannAWS-Region.

Die vorsignierte URL muss eine gültige Anforderung für die `CopyDBClusterSnapshot` API-Aktion sein, die in der Quelle ausgeführt werden kannAWS-Region, die den zu kopierenden Cluster-Snapshot enthält. Die vorsignierte URL-Anforderung muss die folgenden Parameterwerte enthalten:

- `SourceRegion` – Die ID der Region, die den zu kopierenden Snapshot enthält.
- `SourceDBClusterSnapshotIdentifier` – Die Kennung für den zu kopierenden verschlüsselten Cluster-Snapshot. Dieser Bezeichner muss im ARN-Format (Amazon-Ressourcenname) der Quell-AWS-Region angegeben werden. Wenn Sie beispielsweise einen verschlüsselten Cluster-Snapshot aus us-east-1 kopierenAWS-Region, `SourceDBClusterSnapshotIdentifier` sieht Ihr etwa wie folgt aus: `arn:aws:rds:us-east-1:12345678012:sample-cluster:sample-cluster-snapshot`.
- `TargetDBClusterSnapshotIdentifier` – Die Kennung für den zu erstellenden neuen Cluster-Snapshot. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem Cluster-Snapshot zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterSnapshot

Detaillierte Informationen zu einem Cluster-Snapshot.

Typ: DBClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotAlreadyExistsFault

Sie haben bereits einen Cluster-Snapshot mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen -AWS KMSSchlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateDBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen neuen Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Cluster-ID. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster`

Typ: Zeichenfolge

Erforderlich: Ja

Engine

Der Name der Datenbank-Engine, die für diesen Cluster verwendet werden soll.

Zulässige Werte: `docdb`

Typ: Zeichenfolge

Erforderlich: Ja

AvailabilityZonesAvailabilityZone.N

Eine Liste der Amazon EC2 Availability Zones, in denen Instances im DB-Cluster erstellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

BackupRetentionPeriod

Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Sie müssen einen Mindestwert von 1 angeben.

Standard: 1

Einschränkungen:

- Muss ein Wert zwischen 1 und 35 sein.

Typ: Ganzzahl

Erforderlich: Nein

DBClusterParameterGroupName

Der Name der Parameter-Gruppe, die mit diesem Cluster zu verknüpfen ist.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupName

Eine DB-Subnetzgruppe, die diesem DB-Cluster zugeordnet werden soll.

Einschränkungen: Der Wert muss mit dem Namen einer vorhandenen `DBSubnetGroup` übereinstimmen. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn aktiviert `DeletionProtection` ist, kann der Cluster nur gelöscht werden, wenn er geändert und deaktiviert `DeletionProtection` wird. `DeletionProtection` schützt Cluster vor versehentlichem Löschen.

Typ: Boolesch

Erforderlich: Nein

EnableCloudwatchLogsExports.member.N

Eine Liste der Protokolltypen, die für den Export nach Amazon CloudWatch Logs aktiviert werden müssen. Sie können Prüf- oder Profiler-Protokolle aktivieren. Weitere Informationen finden Sie unter Prüfen von Amazon DocumentDB-Ereignissen und Profiling von Amazon DocumentDB-Operationen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

EngineVersion

Die Versionsnummer der zu verwendenden Datenbank-Engine. Die `--engine-version` wird standardmäßig auf die neueste Hauptversion des Moduls festgelegt. Bei Produktions-Workloads empfehlen wir, diesen Parameter explizit mit der beabsichtigten Hauptversion zu deklarieren.

Typ: Zeichenfolge

Erforderlich: Nein

GlobalClusterIdentifier

Die Cluster-ID des neuen globalen Clusters.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

KmsKeyId

Die Kennung des AWS KMS-Schlüssels für einen verschlüsselten Cluster.

Die Kennung für den AWS KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den AWS KMS-Verschlüsselungsschlüssel. Wenn Sie einen DB-Cluster mit demselben AWS-Konto erstellen, das über den AWS KMS-Verschlüsselungsschlüssel verfügt, der für die Verschlüsselung des neuen Clusters verwendet wurde, können Sie den AWS KMS-Schlüssel-Alias anstelle des ARN für den AWS KMS-Verschlüsselungsschlüssel verwenden.

Wenn ein Verschlüsselungsschlüssel in `KmsKeyId` nicht angegeben ist:

- Wenn der `StorageEncrypted`-Parameter `true` lautet, verwendet Amazon DocumentDB Ihren Standard-Verschlüsselungsschlüssel.

AWS KMS erstellt den Standard-Verschlüsselungsschlüssel für Ihr AWS-Konto. Ihr AWS-Konto verfügt für jede AWS-Regionen über einen anderen Standard-Verschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUsername

Der Name des Masterbenutzers für diesen DB-Cluster.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben oder Zahlen enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf kein Wort sein, das für die ausgewählte Datenbank-Engine reserviert ist.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserPassword

Das Passwort für den Masterbenutzer der Datenbank. Dieses Passwort kann alle druckbaren ASCII-Zeichen, außer Schrägstrich (/), doppeltes Anführungszeichen (") oder das "At"-Zeichen (@), enthalten.

Einschränkungen: Muss 8 bis 100 Zeichen enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Portnummer, an dem die Instances im Cluster Verbindungen akzeptieren.

Typ: Ganzzahl

Erforderlich: Nein

PreferredBackupWindow

Der tägliche Zeitraum, in dem automatische Sicherungen erstellt werden, wenn diese mit dem Parameter `BackupRetentionPeriod` aktiviert sind.

Der Standardwert ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitraum pro AWS-Region ausgewählt wird.

Einschränkungen:

- Muss im Format `hh24:mi-hh24:mi` angegeben werden.
- Muss in Universal Coordinated Time (UTC) angegeben werden.
- Darf nicht mit dem bevorzugten Wartungsfenster in Konflikt treten.
- Muss mindestens 30 Minuten betragen.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Der Standardwert ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitraum für jede AWS-Region an einem zufälligen Wochentag ausgewählt wird.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

PreSignedUrl

Wird derzeit nicht unterstützt.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Gibt an, ob der Cluster verschlüsselt ist.

Typ: Boolesch

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB-Entwicklerhandbuch.

Gültige Werte für den Speichertyp – `standard | iopt1`

Der Standardwert ist `standard`

> **Note**
>
> Wenn Sie einen DocumentDB-DB-Cluster mit dem Speichertyp erstellen`iopt1`, wird der Speichertyp in der Antwort zurückgegeben. Der Speichertyp wird nicht zurückgegeben, wenn Sie ihn auf setzen`standard`.

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem Cluster zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

VpcSecurityGroupIdsVpcSecurityGroupId.N

Eine Liste der EC2-VPC-Sicherheitsgruppen, die mit diesem Cluster verknüpft werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName` bezieht sich nicht auf eine vorhandene Cluster-Parametergruppe.

HTTP Status Code: 404

DBClusterQuotaExceededFault

Der Cluster kann nicht erstellt werden, da Sie das maximal zulässige Kontingent für Cluster erreicht haben.

HTTP Status Code: 403

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

GlobalClusterNotFoundFault

bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InsufficientStorageClusterCapacity

Für die aktuelle Aktion ist nicht genügend Speicherplatz verfügbar. Möglicherweise können Sie diesen Fehler beheben, indem Sie Ihre Subnetzgruppe aktualisieren, um verschiedene Availability Zones zu verwenden, in denen mehr Speicher verfügbar ist.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

InvalidDBSubnetGroupStateFault

Die Subnetzgruppe kann nicht gelöscht werden, da sie verwendet wird.

HTTP Status Code: 400

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, während sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nicht alle Availability Zones ab, nachdem sie erstellt wurde, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen -AWS KMSSchlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go

- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateDBClusterParameterGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt eine neue Cluster-Parametergruppe.

Parameter in einer Cluster-Parametergruppe gelten für alle Instances in einem Cluster.

Zu Beginn wird eine Cluster-Parametergruppe mit den Standardparametern für die Datenbank-Engine erstellt, die von den Instances im Cluster verwendet wird. In Amazon DocumentDB können Sie keine Änderungen direkt an der `default.docdb3.6` Cluster-Parametergruppe vornehmen. Wenn Ihr Amazon DocumentDB-Cluster die Standard-Cluster-Parametergruppe verwendet und Sie einen Wert darin ändern möchten, müssen Sie zuerst eine neue Parametergruppe erstellen oder eine vorhandene Parametergruppe kopieren, ändern und dann die geänderte Parametergruppe auf Ihren Cluster anwenden. Damit die neue Cluster-Parametergruppe und die zugehörigen Einstellungen wirksam werden, müssen Sie die Instances im Cluster ohne Failover neu starten. Weitere Informationen finden Sie unter Ändern von Amazon DocumentDB-Cluster-Parametergruppen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der Cluster-Parametergruppe.

Einschränkungen:

- Darf nicht dem Namen einer vorhandenen `DBClusterParameterGroup` entsprechen.

> **Note**
>
> Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Typ: Zeichenfolge

Erforderlich: Ja

DBParameterGroupFamily

Der Name der Cluster-Parametergruppenfamilie.

Typ: Zeichenfolge

Erforderlich: Ja

Description

Die Beschreibung der Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die der Cluster-Parametergruppe zugeordnet werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterParameterGroup

Detaillierte Informationen zu einer Cluster-Parametergruppe.

Typ: DBClusterParameterGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupAlreadyExists

Eine Parametergruppe mit demselben Namen ist bereits vorhanden.

HTTP Status Code: 400

DBParameterGroupQuotaExceeded

Diese Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Parametergruppen überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateDBClusterSnapshot

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen Snapshot eines Clusters.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Kennung des Clusters, für den ein Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Beispiel: `my-cluster`

Typ: Zeichenfolge

Erforderlich: Ja

DBClusterSnapshotIdentifier

Die Kennung des Cluster-Snapshots. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster-snapshot1`

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die dem Cluster-Snapshot zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterSnapshot

Detaillierte Informationen zu einem Cluster-Snapshot.

Typ: DBClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterSnapshotAlreadyExistsFault

Sie haben bereits einen Cluster-Snapshot mit der angegebenen Kennung.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateDBInstance

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt eine neue Instance.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die ID des Clusters, zu dem die Instance gehört.

Typ: Zeichenfolge

Erforderlich: Ja

DBInstanceClass

Die Rechen- und Speicherkapazität der Instance, beispielsweise `db.r5.large`.

Typ: Zeichenfolge

Erforderlich: Ja

DBInstanceIdentifier

Die Instance-ID. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `mydbinstance`

Typ: Zeichenfolge

Erforderlich: Ja

Engine

Der Name der Datenbank-Engine, die für diese Instance verwendet werden soll.

Zulässiger Wert: `docdb`

Typ: Zeichenfolge

Erforderlich: Ja

AutoMinorVersionUpgrade

Dieser Parameter gilt nicht für Amazon DocumentDB. Amazon DocumentDB führt unabhängig vom eingestellten Wert keine kleineren Versions-Upgrades durch.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

AvailabilityZone

Die Amazon EC2-Availability-Zone, in der die Instance erstellt wird.

Standard: Eine zufällig vom System gewählte Availability Zone in der AWS-Region des Endpunkts.

Beispiel: `us-east-1d`

Typ: Zeichenfolge

Erforderlich: Nein

CACertificateIdentifier

Die CA-Zertifikat-ID, die für das Serverzertifikat der DB-Instance verwendet werden soll.

Weitere Informationen finden Sie unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten während der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Typ: Zeichenfolge

Erforderlich: Nein

CopyTagsToSnapshot

Ein Wert, der angibt, ob Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden sollen. Standardmäßig werden Tags nicht kopiert.

Typ: Boolesch

Erforderlich: Nein

EnablePerformanceInsights

Ein Wert, der angibt, ob Performance Insights für die DB-Instance aktiviert werden sollen. Weitere Informationen finden Sie unter Verwenden von Amazon Performance Insights.

Typ: Boolesch

Erforderlich: Nein

PerformanceInsightsKMSKeyId

Die AWS KMS-Schlüssel-ID für die Verschlüsselung von Performance-Insights-Daten.

Die AWS KMS Schlüsselkennung ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel.

Wenn Sie keinen Wert für PerformanceInsightsKMS angebenKeyId, verwendet Amazon DocumentDB Ihren Standard-KMS-Schlüssel. Es gibt einen Standard-KMS-Schlüssel für Ihr Amazon Web Services-Konto. Ihr Amazon-Web-Services-Konto verfügt über einen anderen Standard-KMS-Schlüssel für jede Amazon-Web-Services-Region.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Der Standardwert ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitraum für jede AWS-Region an einem zufälligen Wochentag ausgewählt wird.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance hochgestuft wird.

Standard: 1

Gültige Werte: 0–15

Typ: Ganzzahl

Erforderlich: Nein

Tags.Tag.N

Die Tags, die der Instance zugewiesen werden sollen. Sie können einer Instance bis zu 10 Tags zuweisen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instance.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AuthorizationNotFound

Die angegebene CIDR-IP oder Amazon EC2-Sicherheitsgruppe ist für die angegebene Sicherheitsgruppe nicht autorisiert.

Amazon DocumentDB ist möglicherweise auch nicht autorisiert, mithilfe von IAM die erforderlichen Aktionen in Ihrem Namen durchzuführen.

HTTP Status Code: 404

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceAlreadyExists

Sie haben bereits eine Instance mit der angegebenen Kennung.

HTTP Status Code: 400

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

DBSecurityGroupNotFound

`DBSecurityGroupName` bezieht sich nicht auf eine vorhandene Sicherheitsgruppe.

HTTP Status Code: 404

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

InstanceQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Instances überschreiten.

HTTP Status Code: 400

InsufficientDBInstanceCapacity

Die angegebene Instance-Klasse ist in der angegebenen Availability Zone nicht verfügbar.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nicht alle Availability Zones ab, nachdem sie erstellt wurde, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen -AWS KMSSchlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

StorageTypeNotSupported

Der angegebene Speicher `StorageType` kann nicht mit der DB-Instance verknüpft werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateDBSubnetGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt eine neue Subnetzgruppe. Subnetzgruppen müssen mindestens ein Subnetz in mindestens zwei Availability Zones in der enthaltenAWS-Region.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupDescription

Die Beschreibung für die Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Ja

DBSubnetGroupName

Der Name für die Subnetzgruppe. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen: Darf nicht mehr als 255 Buchstaben, Ziffern, Punkte, Unterstriche, Leerzeichen und Bindestriche enthalten. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Ja

SubnetIdsSubnetIdentifier.N

Die Amazon EC2-Subnetz-IDs für die Subnetzgruppe.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

Tags.Tag.N

Das Tag, das der Subnetzgruppe zugeordnet werden soll.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBSubnetGroup

Detaillierte Informationen zu einer Subnetzgruppe.

Typ: DBSubnetGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupAlreadyExists

`DBSubnetGroupName` wird bereits von einer vorhandenen Subnetzgruppe verwendet.

HTTP Status Code: 400

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Subnetzgruppen überschreiten.

HTTP Status Code: 400

DBSubnetQuotaExceededFault

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Subnetzen in einer Subnetzgruppe überschreiten.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateEventSubscription

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt ein Abonnement für Amazon DocumentDB-Ereignisbenachrichtigungen. Für diese Aktion ist ein Amazon-Ressourcenname (ARN) des Themas erforderlich, der mithilfe der Amazon DocumentDB-Konsole, der Amazon SNS-Konsole oder der Amazon SNS-API erstellt wurde. Um einen ARN mit Amazon SNS zu erhalten, müssen Sie ein Thema in Amazon SNS erstellen und das Thema abonnieren. Der ARN wird in der Amazon SNS-Konsole angezeigt.

Sie können den Typ der Quelle (`SourceType`) angeben, über die Sie benachrichtigt werden möchten. Sie können auch eine Liste der Amazon DocumentDB-Quellen (`SourceIds`) bereitstellen, die die Ereignisse auslösen, und Sie können eine Liste der Ereigniskategorien (`EventCategories`) für Ereignisse bereitstellen, über die Sie benachrichtigt werden möchten. Sie können beispielsweise und angeben`SourceType = db-instanceSourceIds = mydbinstance1, mydbinstance2EventCategories = Availability, Backup`.

Wenn Sie sowohl als auch angeben `SourceType` `SourceIds` (z. B. `SourceType = db-instance` und `SourceIdentifier = myDBInstance1`), werden Sie über alle `db-instance` Ereignisse für die angegebene Quelle benachrichtigt. Wenn Sie einen angeben`SourceType`, aber keinen angeben`SourceIdentifier`, erhalten Sie eine Benachrichtigung über die Ereignisse für diesen Quelltyp für alle Ihre Amazon DocumentDB-Quellen. Wenn Sie weder die noch die angeben`SourceIdentifier`, werden Sie über Ereignisse informiert, `SourceType` die aus allen Amazon DocumentDB-Quellen generiert werden, die zu Ihrem Kundenkonto gehören.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SnsTopicArn

Der Amazon-Ressourcenname (ARN) des SNS-Themas, das für die Ereignisbenachrichtigung erstellt wurde. Amazon SNS erstellt den ARN, wenn Sie ein Thema erstellen und es abonnieren.

Typ: Zeichenfolge

Erforderlich: Ja

SubscriptionName

Der Name des Abonnements.

Einschränkungen: Der Name muss weniger als 255 Zeichen lang sein.

Typ: Zeichenfolge

Erforderlich: Ja

Enabled

Ein boolescher Wert; setzen Sie ihn auf , `true` um das Abonnement zu aktivieren, setzen Sie ihn auf , um das Abonnement `false` zu erstellen, aber nicht zu aktivieren.

Typ: Boolesch

Erforderlich: Nein

EventCategoriesEventCategory.N

Eine Liste von Ereigniskategorien für einen `SourceType`, den Sie abonnieren möchten.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SourceIdsSourceId.N

Die Liste der IDs der Ereignisquellen, für die Ereignisse zurückgegeben werden. Wenn nicht angegeben, werden alle Quellen zur Antwort hinzugefügt. Eine ID muss mit einem Buchstaben beginnen und darf nur ASCII-Buchstaben, Ziffern und Bindestriche enthalten; sie darf nicht mit einem Bindestrich oder zwei aufeinander folgenden Bindestrichen enden.

Einschränkungen:

- Falls `SourceIds` angegeben, `SourceType` muss auch angegeben werden.
- Wenn der Quelltyp eine Instance ist, `DBInstanceIdentifier` muss ein angegeben werden.
- Wenn der Quelltyp eine Sicherheitsgruppe ist, `DBSecurityGroupName` muss ein angegeben werden.
- Wenn der Quelltyp eine Parametergruppe ist, `DBParameterGroupName` muss ein angegeben werden.
- Wenn der Quelltyp ein Snapshot ist, `DBSnapshotIdentifier` muss ein angegeben werden.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SourceType

Der Typ der Quelle, die die Ereignisse generiert. Wenn Sie beispielsweise über Ereignisse benachrichtigt werden möchten, die von einer Instance generiert werden, würden Sie diesen Parameter auf setzen`db-instance`. Wenn der Wert nicht angegeben ist, werden alle Ereignisse zurückgegeben.

Gültige Werte: `db-instance`, `db-cluster`, `db-parameter-group`, `db-security-group`, `db-cluster-snapshot`

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem Ereignisabonnement zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einem Ereignis, das Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

EventSubscriptionQuotaExceeded

Sie haben die maximale Anzahl von Ereignisabonnements erreicht.

HTTP Status Code: 400

SNSInvalidTopic

Amazon SNS hat darauf geantwortet, dass ein Problem mit dem angegebenen Thema vorliegt.

HTTP Status Code: 400

SNSNoAuthorization

Sie sind nicht berechtigt, im Amazon-Ressourcennamen (ARN) des SNS-Themas zu veröffentlichen.

HTTP Status Code: 400

SNSTopicArnNotFound

Der Amazon-Ressourcenname (ARN) des SNS-Themas ist nicht vorhanden.

HTTP Status Code: 404

SourceNotFound

Die angeforderte Quelle konnte nicht gefunden werden.

HTTP Status Code: 404

SubscriptionAlreadyExist

Der angegebene Abonnementname ist bereits vorhanden.

HTTP Status Code: 400

SubscriptionCategoryNotFound

Die angegebene Kategorie ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++

- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

CreateGlobalCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen globalen Amazon DocumentDB-Cluster, der sich über mehrere erstrecken kannAWS-Regionen. Der globale Cluster enthält einen primären Cluster mit Lese-/Schreibfähigkeit und bis zu schreibgeschützten sekundären Clustern. Globale Cluster verwenden eine speicherbasierte schnelle Replikation über Regionen hinweg mit Latenzen von weniger als einer Sekunde, wobei eine dedizierte Infrastruktur verwendet wird, ohne die Leistung Ihres Workloads zu beeinträchtigen.

Sie können einen globalen Cluster erstellen, der zunächst leer ist, und ihm dann einen primären und einen sekundären Cluster hinzufügen. Oder Sie können während der Erstellung einen vorhandenen Cluster angeben, und dieser Cluster wird zum primären Cluster des globalen Clusters.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die Cluster-ID des neuen globalen Clusters.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

DatabaseName

Der Name für Ihre Datenbank mit bis zu 64 alphanumerischen Zeichen. Wenn Sie keinen Namen angeben, erstellt Amazon DocumentDB keine Datenbank in dem globalen Cluster, den Sie erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Die Löschschutzeinstellung für den neuen globalen Cluster. Der globale Cluster kann nicht gelöscht werden, wenn der Löschschutz aktiviert ist.

Typ: Boolesch

Erforderlich: Nein

Engine

Der Name der Datenbank-Engine, die für diesen Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die Engine-Version des globalen Clusters.

Typ: Zeichenfolge

Erforderlich: Nein

SourceDBClusterIdentifier

Der Amazon-Ressourcenname (ARN), der als primärer Cluster des globalen Clusters verwendet werden soll. Dieser Parameter ist optional.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Die Speicherverschlüsselungseinstellung für den neuen globalen Cluster.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

GlobalClusterAlreadyExistsFault

`GlobalClusterIdentifier` ist bereits vorhanden. Wählen Sie eine neue globale Cluster-ID (eindeutiger Name), um einen neuen globalen Cluster zu erstellen.

HTTP Status Code: 400

GlobalClusterQuotaExceededFault

Die Anzahl der globalen Cluster für dieses Konto ist bereits auf dem maximal zulässigen Wert.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteDBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht einen zuvor bereitgestellten Cluster. Wenn Sie einen Cluster löschen, werden alle automatisierten Backups für diesen Cluster gelöscht und können nicht wiederhergestellt werden. Manuelle DB-Cluster-Snapshots des angegebenen Clusters werden nicht gelöscht.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Cluster-ID für den zu löschenden Cluster. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit einem vorhandenen übereinstimmen`DBClusterIdentifier`.

Typ: Zeichenfolge

Erforderlich: Ja

FinalDBSnapshotIdentifier

Die Cluster-Snapshot-ID des neuen Cluster-Snapshots, der erstellt wurde, wenn auf gesetzt `SkipFinalSnapshot` ist`false`.

> **Note**
>
> Die Angabe dieses Parameters und die Festlegung des `SkipFinalShapshot` Parameters auf `true` führen zu einem Fehler.

Einschränkungen:

- Muss zwischen 1 und 255 Buchstaben, Zahlen oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

SkipFinalSnapshot

Bestimmt, ob ein endgültiger Cluster-Snapshot erstellt wird, bevor der Cluster gelöscht wird. Wenn angegeben `true` ist, wird kein Cluster-Snapshot erstellt. Wenn angegeben `false` ist, wird ein Cluster-Snapshot erstellt, bevor der DB-Cluster gelöscht wird.

> **Note**
>
> Wenn `SkipFinalSnapshot` ist`false`, müssen Sie einen `FinalDBSnapshotIdentifier` Parameter angeben.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterSnapshotAlreadyExistsFault

Sie haben bereits einen Cluster-Snapshot mit der angegebenen Kennung.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteDBClusterParameterGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht eine angegebene Cluster-Parametergruppe. Die zu löschende Cluster-Parametergruppe kann keinem Cluster zugeordnet werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der Cluster-Parametergruppe.

Einschränkungen:

- Muss der Name einer vorhandenen Cluster-Parametergruppe sein.
- Sie können eine Standard-Cluster-Parametergruppe nicht löschen.
- Kann keinem Cluster zugeordnet werden.

Typ: Zeichenfolge

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

InvalidDBParameterGroupState

Die Parametergruppe wird verwendet oder befindet sich in einem ungültigen Zustand. Wenn Sie versuchen, die Parametergruppe zu löschen, können Sie sie nicht löschen, wenn sich die Parametergruppe in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteDBClusterSnapshot

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht einen Cluster-Snapshot. Wenn der Snapshot gerade kopiert wird, wird der Kopiervorgang beendet.

> **Note**
>
> Der Cluster-Snapshot muss sich im `available` Status befinden, um gelöscht werden zu können.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterSnapshotIdentifier

Die Kennung des zu löschenden Cluster-Snapshots.

Einschränkungen: Muss der Name eines vorhandenen Cluster-Snapshots im `available` Status sein.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterSnapshot

Detaillierte Informationen zu einem Cluster-Snapshot.

Typ: DBClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteDBInstance

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht eine zuvor bereitgestellte Instance.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die Instance-ID für die zu löschende Instance. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit dem Namen einer vorhandenen Instance übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instance.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

DBSnapshotAlreadyExists

`DBSnapshotIdentifier` wird bereits von einem vorhandenen Snapshot verwendet.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteDBSubnetGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht eine Subnetzgruppe.

> **Note**
>
> Die angegebene Datenbanksubnetzgruppe muss nicht mit beliebigen DB-Instances verknüpft werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupName

Der Name der zu löschenden Datenbanksubnetzgruppe.

> **Note**
>
> Sie können die Standardsubnetzgruppe nicht löschen.

Einschränkungen:

Muss dem Namen einer vorhandenen `DBSubnetGroup` entsprechen. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

InvalidDBSubnetGroupStateFault

Die Subnetzgruppe kann nicht gelöscht werden, da sie verwendet wird.

HTTP Status Code: 400

InvalidDBSubnetStateFault

Das Subnetz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteEventSubscription

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht ein Abonnement für Amazon DocumentDB-Ereignisbenachrichtigungen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SubscriptionName

Der Name des Abonnements für Amazon DocumentDB-Ereignisbenachrichtigungen, das Sie löschen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einem Ereignis, das Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InvalidEventSubscriptionState

Möglicherweise ändert jemand anderes ein Abonnement. Warten Sie einige Sekunden und versuchen Sie es erneut.

HTTP Status Code: 400

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DeleteGlobalCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Löscht einen globalen Cluster. Die primären und sekundären Cluster müssen bereits getrennt oder gelöscht werden, bevor Sie versuchen, einen globalen Cluster zu löschen.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die Cluster-ID des globalen Clusters, der gelöscht wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

GlobalClusterNotFoundFault

bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, während sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeCertificates

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste der Zertifizierungsstellenzertifikate (CA) zurück, die von Amazon DocumentDB für dieses bereitgestellt werdenAWS-Konto.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

CertificateIdentifier

Die vom Benutzer bereitgestellte Zertifikat-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen für das angegebene Zertifikat zurückgegeben. Wenn dieser Parameter weggelassen wird, wird eine Liste von bis zu `MaxRecords` Zertifikaten zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beschränkungen

- Muss mit einem vorhandenen übereinstimmen`CertificateIdentifier`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen `DescribeCertificates`-Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehrere Datensätze vorhanden sind, als der Wert `MaxRecords` angibt, ist ein Paginierungstoken mit dem Namen einer Markierung in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen:

- Minimum: 20
- Maximum: 100

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

Zertifikate.Zertifikat.N

Eine Liste von Zertifikaten für diese AWS-Konto.

Typ: Array von Certificate-Objekten

Marker

Ein optionales Paginierungstoken, das bereitgestellt wird, wenn die Anzahl der abgerufenen Datensätze größer als ist`MaxRecords`. Wenn dieser Parameter angegeben ist, gibt die Markierung den nächsten Datensatz in der Liste an. Wenn Sie den Wert von `Marker` im nächsten Aufruf von angeben, `DescribeCertificates` wird die nächste Seite mit Zertifikaten angezeigt.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

CertificateNotFound

`CertificateIdentifier` bezieht sich nicht auf ein vorhandenes Zertifikat.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBClusterParameterGroups

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von `DBClusterParameterGroup`-Beschreibungen zurück. Wenn ein `DBClusterParameterGroupName` Parameter angegeben wird, enthält die Liste nur die Beschreibung der angegebenen Cluster-Parametergruppe.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name einer bestimmten Cluster-Parametergruppe, für die Details zurückgegeben werden sollen.

Einschränkungen:

- Falls angegeben, muss mit dem Namen einer vorhandenen übereinstimmen`DBClusterParameterGroup`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

DB ClusterParameterGroups.DB ClusterParameterGroup.N

Eine Liste von Cluster-Parametergruppen.

Typ: Array von DBClusterParameterGroup-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBClusterParameters

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt die detaillierte Parameterliste für eine bestimmte Cluster-Parametergruppe zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name einer bestimmten Cluster-Parametergruppe, für die Parameterdetails zurückgegeben werden sollen.

Einschränkungen:

- Falls angegeben, muss mit dem Namen einer vorhandenen übereinstimmenDBClusterParameterGroup.

Typ: Zeichenfolge

Erforderlich: Ja

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Source

Dieser Wert gibt an, dass nur die Parameter für eine bestimmte Quelle zurückgegeben werden sollen. Parameterquellen können `engine`, `service` oder `customer` sein.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Parameter.Parameter.N

Gibt eine Liste der Parameter für die Cluster-Parametergruppe zurück.

Typ: Array von Parameter-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBClusters

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen zu bereitgestellten Amazon DocumentDB-Clustern zurück. Diese API-Operation unterstützt die Paginierung. Für bestimmte Verwaltungsfunktionen wie das Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS und Amazon Neptune geteilt wird. Verwenden Sie den `filterName=engine,Values=docdb` Filterparameter, um nur Amazon DocumentDB-Cluster zurückzugeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die vom Benutzer bereitgestellte Cluster-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen aus dem spezifischen Cluster zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Falls angegeben, muss mit einem vorhandenen übereinstimmen`DBClusterIdentifier`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Ein Filter, der einen oder mehrere zu beschreibende Cluster angibt.

Unterstützte Filter:

- `db-cluster-id` : Akzeptiert Cluster-IDs und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur Informationen zu den Clustern, die durch diese ARNs identifiziert werden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

DBClusters.DBCluster.N

Eine Liste von Clustern.

Typ: Array von DBCluster-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBClusterSnapshotAttributes

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste der Namen und Werte von Cluster-Snapshot-Attributen für einen manuellen DB-Cluster-Snapshot zurück.

Wenn Sie Snapshots für andere freigebenAWS-Konten, `DescribeDBClusterSnapshotAttributes` gibt das `restore` Attribut und eine Liste von IDs für die zurückAWS-Konten, die zum Kopieren oder Wiederherstellen des manuellen Cluster-Snapshots autorisiert sind. Wenn in der Liste der Werte für das `restore` Attribut enthalten `all` ist, ist der manuelle Cluster-Snapshot öffentlich und kann von allen kopiert oder wiederhergestellt werdenAWS-Konten.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterSnapshotIdentifier

Die Kennung für den Cluster-Snapshot, für den die Attribute beschrieben werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterSnapshotAttributesResult

Detaillierte Informationen zu den Attributen, die einem Cluster-Snapshot zugeordnet sind.

Typ: DBClusterSnapshotAttributesResult Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBClusterSnapshots

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen zu Cluster-Snapshots zurück. Diese API-Operation unterstützt die Paginierung.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die ID des Clusters, für den die Liste der Cluster-Snapshots abgerufen werden soll. Dieser Parameter kann nicht mit dem `DBClusterSnapshotIdentifier` Parameter verwendet werden. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Falls angegeben, muss mit der Kennung eines vorhandenen übereinstimmen`DBCluster`.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterSnapshotIdentifier

Eine spezifische zu beschreibende Cluster-Snapshot-Kennung. Dieser Parameter kann nicht mit dem `DBClusterIdentifier` Parameter verwendet werden. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Falls angegeben, muss mit der Kennung eines vorhandenen übereinstimmen`DBClusterSnapshot`.
- Wenn diese Kennung für einen automatisierten Snapshot ist, muss auch der Parameter `SnapshotType` angegeben werden.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

IncludePublic

Legen Sie den Wert auf fest, `true` um manuelle Cluster-Snapshots einzuschließen, die öffentlich sind und von jedem kopiert oder wiederhergestellt werden könnenAWS-Konto, andernfalls `false`. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

IncludeShared

Legen Sie den Wert auf fest, `true` um freigegebene manuelle Cluster-Snapshots von anderen einzuschließenAWS-Konten, denen die Berechtigung zum Kopieren oder Wiederherstellen erteilt AWS-Konto wurde, andernfalls `false`. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

SnapshotType

Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:

- `automated` – Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr erstellt hatAWS-Konto.
- `manual` – Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr erstellt habenAWS-Konto.
- `shared` – Gibt alle manuellen Cluster-Snapshots zurück, die für Ihr freigegeben wurdenAWS-Konto.
- `public` – Gibt alle Cluster-Snapshots zurück, die als öffentlich markiert wurden.

Wenn Sie keinen `SnapshotType` Wert angeben, werden sowohl automatisierte als auch manuelle Cluster-Snapshots zurückgegeben. Sie können freigegebene Cluster-Snapshots in diese Ergebnisse aufnehmen, indem Sie den `IncludeShared` Parameter auf setzen`true`. Sie können öffentliche Cluster-Snapshots in diese Ergebnisse aufnehmen, indem Sie den`IncludePublic` Parameter auf setzen`true`.

Die Parameter `IncludePublic` und `IncludeShared` gelten nicht für `SnapshotType`-Werte von `manual` oder `automated`. Der Parameter `IncludePublic` gilt nicht, wenn `SnapshotType` auf `shared` festgelegt ist. Der Parameter `IncludeShared` gilt nicht, wenn `SnapshotType` auf `public` festgelegt ist.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

DB ClusterSnapshots.DB ClusterSnapshot.N

Stellt eine Liste von Cluster-Snapshots bereit.

Typ: Array von DBClusterSnapshot-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBEngineVersions

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste der verfügbaren Engines zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBParameterGroupFamily

Der Name einer bestimmten Parametergruppenfamilie, für die Details zurückgegeben werden sollen.

Einschränkungen:

- Falls angegeben, muss mit einem vorhandenen übereinstimmen`DBParameterGroupFamily`.

Typ: Zeichenfolge

Erforderlich: Nein

DefaultOnly

Gibt an, dass nur die standardmäßige Version der angegebenen Engine oder Kombination aus Engine und Hauptversion zurückgegeben wird.

Typ: Boolesch

Erforderlich: Nein

Engine

Die zurückzugebende Datenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die zurückzugebende Datenbank-Engine-Version.

Beispiel: `3.6.0`

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

ListSupportedCharacterSets

Wenn dieser Parameter angegeben ist und die angeforderte Engine den Parameter `CharacterSetName` für `CreateDBInstance` unterstützt, enthält die Antwort eine Liste der unterstützten Zeichensätze für jede Engine-Version.

Typ: Boolesch

Erforderlich: Nein

ListSupportedTimezones

Wenn dieser Parameter angegeben ist und die angeforderte Engine den Parameter `TimeZone` für `CreateDBInstance` unterstützt, enthält die Antwort eine Liste der unterstützten Zeitzonen für jede Engine-Version.

Typ: Boolesch

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

DB EngineVersions.DB EngineVersion.N

Detaillierte Informationen zu einer oder mehreren Engine-Versionen.

Typ: Array von DBEngineVersion-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2

- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBInstances

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen zu bereitgestellten Amazon DocumentDB-Instances zurück. Diese API unterstützt Paginierung.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die vom Benutzer bereitgestellte Instance-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen von der spezifischen Instance zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Falls angegeben, muss mit der Kennung eines vorhandenen übereinstimmen`DBInstance`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Ein Filter, der eine oder mehrere zu beschreibende Instances angibt.

Unterstützte Filter:

- `db-cluster-id` : Akzeptiert Cluster-IDs und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur die Informationen zu den Instances, die den Clustern zugeordnet sind, die durch diese ARNs identifiziert werden.
- `db-instance-id` : Akzeptiert Instance-IDs und Instance-ARNs. Die Ergebnisliste enthält nur die Informationen zu den Instances, die durch diese ARNs identifiziert werden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

DBInstances.DBInstance.N

Detaillierte Informationen über eine oder mehrere Instances.

Typ: Array von DBInstance-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeDBSubnetGroups

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von `DBSubnetGroup`-Beschreibungen zurück. Wenn eine angegeben `DBSubnetGroupName` ist, enthält die Liste nur die Beschreibungen der angegebenen `DBSubnetGroup`.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupName

Der Name der Subnetzgruppe, für die Details zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

DB SubnetGroups.DB SubnetGroup.N

Detaillierte Informationen über eine oder mehrere Subnetzgruppen.

Typ: Array von DBSubnetGroup-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeEngineDefaultClusterParameters

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt die Standard-Engine- und System-Parameterinformationen für die Cluster-Datenbank-Engine zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBParameterGroupFamily

Der Name der Cluster-Parametergruppenfamilie, für die die Engine-Parameterinformationen zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EngineDefaults

Enthält das Ergebnis eines erfolgreichen Aufrufs der `DescribeEngineDefaultClusterParameters` Operation.

Typ: EngineDefaults Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeEventCategories

Service: Amazon DocumentDB (with MongoDB compatibility)

Zeigt eine Liste der Kategorien für alle Ereignisquelltypen oder – falls angegeben – für einen angegebenen Quelltyp an.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

SourceType

Der Typ der Quelle, die die Ereignisse generiert.

Zulässige Werte: `db-instance`, `db-parameter-group`, `db-security-group`

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventCategoriesMapList

Eine Liste von Ereigniskategoriezuordnungen.

Typ: Array von EventCategoriesMap-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeEvents

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt Ereignisse im Zusammenhang mit Instances, Sicherheitsgruppen, Snapshots und DB-Parametergruppen der letzten 14 Tage zurück. Sie können Ereignisse abrufen, die für eine bestimmte DB-Instance, Sicherheitsgruppe, einen Snapshot oder eine Parametergruppe spezifisch sind, indem Sie den Namen als Parameter angeben. Standardmäßig werden die Ereignisse der letzten Stunde zurückgegeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Duration

Die Anzahl der Minuten, in denen Ereignisse abgerufen werden sollen.

Standard: 60

Typ: Ganzzahl

Erforderlich: Nein

EndTime

Das Ende des Zeitintervalls, für das Ereignisse abgerufen werden sollen, angegeben im ISO 8601-Format.

Beispiel: 2009-07-08T18:00Z

Typ: Zeitstempel

Erforderlich: Nein

EventCategoriesEventCategory.N

Eine Liste von Ereigniskategorien, die Benachrichtigungen für ein Abonnement für Ereignisbenachrichtigungen auslösen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

SourceIdentifier

ID der Ereignisquelle, für die Ereignisse zurückgegeben werden. Wenn nicht angegeben, werden alle Quellen zur Antwort hinzugefügt.

Einschränkungen:

- Wenn angegeben `SourceIdentifier` ist, `SourceType` muss auch angegeben werden.
- Wenn der Quelltyp ist`DBInstance`, `DBInstanceIdentifier` muss ein angegeben werden.
- Wenn der Quelltyp ist`DBSecurityGroup`, `DBSecurityGroupName` muss ein angegeben werden.

- Wenn der Quelltyp ist`DBParameterGroup`, `DBParameterGroupName` muss ein angegeben werden.
- Wenn der Quelltyp ist`DBSnapshot`, `DBSnapshotIdentifier` muss ein angegeben werden.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

SourceType

Die Ereignisquelle zum Abrufen von Ereignissen. Wenn kein Wert angegeben ist, werden alle Ereignisse zurückgegeben.

Typ: Zeichenfolge

Zulässige Werte: `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Erforderlich: Nein

StartTime

Der Beginn des Zeitintervalls, für das Ereignisse abgerufen werden sollen, angegeben im ISO 8601-Format.

Beispiel: 2009-07-08T18:00Z

Typ: Zeitstempel

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

Events.Event.N

Detaillierte Informationen zu einem oder mehreren Ereignissen.

Typ: Array von Event-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeEventSubscriptions

Service: Amazon DocumentDB (with MongoDB compatibility)

Listet alle Abonnementbeschreibungen für ein Kundenkonto auf. Die Beschreibung für ein Abonnement umfasst `SubscriptionName`, `SNSTopicARN`, `CustomerID`, `SourceType`, `SourceID`, `CreationTime`, und `Status`.

Wenn Sie einen angeben`SubscriptionName`, listet die Beschreibung für dieses Abonnement auf.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

SubscriptionName

Der Name des Abonnements für Amazon DocumentDB-Ereignisbenachrichtigungen, das Sie beschreiben möchten.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

EventSubscriptionsListEventSubscription.N

Eine Liste von Ereignisabonnements.

Typ: Array von EventSubscription-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeGlobalClusters

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen zu globalen Amazon DocumentDB-Clustern zurück. Diese API unterstützt Paginierung.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Ein Filter, der einen oder mehrere zu beschreibende globale DB-Cluster angibt.

Unterstützte Filter: `db-cluster-id` akzeptiert Cluster-IDs und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur Informationen über die durch diese ARNs identifizierten Cluster.

Typ: Array von Filter-Objekten

Erforderlich: Nein

GlobalClusterIdentifier

Die vom Benutzer bereitgestellte Cluster-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen aus dem spezifischen Cluster zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen `DescribeGlobalClusters`-Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze vorhanden sind als der angegebene `MaxRecords` Wert, wird ein Paginierungstoken, eine sogenannte Markierung, in die Antwort aufgenommen, sodass Sie die verbleibenden Ergebnisse abrufen können.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

GlobalClustersGlobalClusterMember.N

Typ: Array von GlobalCluster-Objekten

Marker

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

GlobalClusterNotFoundFault

bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribeOrderableDBInstanceOptions

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste der bestellbaren Instance-Optionen für die angegebene Engine zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Engine

Der Name der Engine, für die Instance-Optionen abgerufen werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

DBInstanceClass

Der Wert des Instance-Klassenfilters. Geben Sie diesen Parameter an, um nur die verfügbaren Angebote anzuzeigen, die der angegebenen Instance-Klasse entsprechen.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Der Filterwert der Engine-Version. Geben Sie diesen Parameter an, um nur die verfügbaren Angebote anzuzeigen, die der angegebenen Engine-Version entsprechen.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

LicenseModel

Der Filterwert des Lizenzmodells. Geben Sie diesen Parameter an, um nur die verfügbaren Angebote anzuzeigen, die dem angegebenen Lizenzmodell entsprechen.

Typ: Zeichenfolge

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Vpc

Der Wert des Virtual Private Cloud (VPC)-Filters. Geben Sie diesen Parameter an, um nur die verfügbaren VPC- oder Nicht-VPC-Angebote anzuzeigen.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

OrderableDBInstanceOptions .OrderableDBInstanceOption .N

Die Optionen, die für eine bestimmte bestellbare Instance verfügbar sind.

Typ: Array von OrderableDBInstanceOption-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

DescribePendingMaintenanceActions

Service: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste der Ressourcen (z. B. Instances) zurück, für die mindestens eine Wartungsaktion aussteht.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Ein Filter, der eine oder mehrere Ressourcen angibt, für die ausstehende Wartungsaktionen zurückgegeben werden sollen.

Unterstützte Filter:

- `db-cluster-id` : Akzeptiert Cluster-IDs und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur ausstehende Wartungsaktionen für die durch diese ARNs identifizierten Cluster.
- `db-instance-id` : Akzeptiert Instance-IDs und Instance-ARNs. Die Ergebnisliste enthält nur ausstehende Wartungsaktionen für die DB-Instances, die durch diese ARNs identifiziert werden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, damit die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

ResourceIdentifier

Der ARN einer Ressource, für die ausstehende Wartungsaktionen zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

PendingMaintenanceActionsResourcePendingMaintenanceActions.N

Die Wartungsaktionen, die angewendet werden sollen.

Typ: Array von ResourcePendingMaintenanceActions-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

ResourceNotFoundFault

Die angegebene Ressourcen-ID wurde nicht gefunden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

FailoverDBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Erzwingt ein Failover für einen Cluster.

Ein Failover für einen Cluster stuft eines der Amazon DocumentDB-Replikate (schreibgeschützte Instances) im Cluster zur primären Instance (Cluster-Writer) hoch.

Wenn die primäre Instance ausfällt, führt Amazon DocumentDB automatisch ein Failover auf ein Amazon DocumentDB-Replikat durch, falls eines vorhanden ist. Sie können ein Failover erzwingen, wenn Sie einen Ausfall einer primären Instance zum Testen simulieren möchten.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Eine Cluster-ID, für die ein Failover erzwungen werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Nein

TargetDBInstanceIdentifier

Der Name der Instance, die zur primären Instance hochgestuft werden soll.

Sie müssen die Instance-ID für ein Amazon DocumentDB-Replikat im Cluster angeben. Beispiel, `mydbcluster-replica1`.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3

- AWS-SDK für Python
- AWS-SDK für Ruby V3

ListTagsForResource

Service: Amazon DocumentDB (with MongoDB compatibility)

Listet alle Tags auf einer Amazon DocumentDB-Ressource auf.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ResourceName

Die Amazon DocumentDB-Ressource mit aufzulistenden Tags. Dieser Wert ist ein Amazon-Ressourcenname (ARN).

Typ: Zeichenfolge

Erforderlich: Ja

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

TagList.Tag.N

Eine Liste mit einem oder mehreren Tags.

Typ: Array von Tag-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ModifyDBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändert eine Einstellung für einen Amazon DocumentDB-Cluster. Sie können einen oder mehrere Datenbank-Konfigurationsparameter ändern, indem Sie diese Parameter und die neuen Werte in der Anforderung angeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Cluster-ID für den Cluster, der geändert wird. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

AllowMajorVersionUpgrade

Ein Wert, der angibt, ob Major-Versionsupgrades erlaubt sind.

Einschränkungen: Sie müssen Hauptversions-Upgrades zulassen, wenn Sie einen Wert für den `EngineVersion` Parameter angeben, der sich von der aktuellen Version des DB-Clusters unterscheidet.

Typ: Boolesch

Erforderlich: Nein

ApplyImmediately

Ein Wert, der angibt, ob die Änderungen in dieser Anforderung und alle ausstehenden Änderungen schnellstmöglich asynchron angewendet werden, unabhängig von der `PreferredMaintenanceWindow` Einstellung für den Cluster. Wenn dieser Parameter auf gesetzt ist`false`, werden Änderungen am Cluster während des nächsten Wartungsfensters übernommen.

Der `ApplyImmediately` Parameter wirkt sich nur auf die `MasterUserPassword` Werte `NewDBClusterIdentifier` und aus. Wenn Sie diesen Parameterwert auf setzen`false`, werden die Änderungen an den `MasterUserPassword` Werten `NewDBClusterIdentifier` und während des nächsten Wartungsfensters angewendet. Alle anderen Änderungen werden sofort übernommen, unabhängig von dem Wert des Parameters `ApplyImmediately`.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

BackupRetentionPeriod

Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Sie müssen einen Mindestwert von 1 angeben.

Standard: 1

Einschränkungen:

- Muss ein Wert zwischen 1 und 35 sein.

Typ: Ganzzahl

Erforderlich: Nein

CloudwatchLogsExportConfiguration

Die Konfigurationseinstellung für die Protokolltypen, die für den Export nach Amazon CloudWatch Logs für eine bestimmte Instance oder einen bestimmten Cluster aktiviert werden sollen. Die `DisableLogTypes`Arrays `EnableLogTypes` und bestimmen, welche Protokolle in CloudWatch Protokolle exportiert (oder nicht exportiert) werden.

Typ: CloudwatchLogsExportConfiguration Objekt

Erforderlich: Nein

DBClusterParameterGroupName

Der Name der Cluster-Parametergruppe, die für den Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn aktiviert `DeletionProtection` ist, kann der Cluster nur gelöscht werden, wenn er geändert und deaktiviert `DeletionProtection` wurde. `DeletionProtection` schützt Cluster vor versehentlichem Löschen.

Typ: Boolesch

Erforderlich: Nein

EngineVersion

Die Versionsnummer der Datenbank-Engine, auf die ein Upgrade durchgeführt werden soll. Das Ändern dieses Parameters führt zu einem Nutzungsausfall. Die Änderung wird im nächsten Wartungsfenster übernommen, es sei denn, `ApplyImmediately` ist aktiviert.

Verwenden Sie den folgenden Befehl, um alle verfügbaren Engine-Versionen für Amazon DocumentDB aufzulisten:

```
aws docdb describe-db-engine-versions --engine docdb --query
"DBEngineVersions[].EngineVersion"
```

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserPassword

Das Passwort für den Masterbenutzer der Datenbank. Dieses Passwort kann alle druckbaren ASCII-Zeichen, außer Schrägstrich (/), doppeltes Anführungszeichen (") oder das "At"-Zeichen (@), enthalten.

Einschränkungen: Muss 8 bis 100 Zeichen enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

NewDBClusterIdentifier

Die neue Cluster-ID für den Cluster beim Umbenennen eines Clusters. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster2`

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Nummer des Ports, an dem der Cluster Verbindungen akzeptiert.

Einschränkungen: Muss ein Wert von `1150` bis sein`65535`.

Standard: Derselbe Port wie der ursprüngliche Cluster.

Typ: Ganzzahl

Erforderlich: Nein

PreferredBackupWindow

Der tägliche Zeitraum, in dem automatische Sicherungen erstellt werden, wenn diese mit dem Parameter `BackupRetentionPeriod` aktiviert sind.

Der Standardwert ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitraum pro AWS-Region ausgewählt wird.

Einschränkungen:

- Muss im Format `hh24:mi-hh24:mi` angegeben werden.
- Muss in Universal Coordinated Time (UTC) angegeben werden.
- Darf nicht mit dem bevorzugten Wartungsfenster in Konflikt treten.
- Muss mindestens 30 Minuten betragen.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Der Standardwert ist ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitraum für jede AWS-Region an einem zufälligen Wochentag ausgewählt wird.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB-Entwicklerhandbuch.

Gültige Werte für den Speichertyp – `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

VpcSecurityGroupIdsVpcSecurityGroupId.N

Eine Liste der Virtual Private Cloud (VPC)-Sicherheitsgruppen, zu denen der Cluster gehören wird.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName` bezieht sich nicht auf eine vorhandene Cluster-Parametergruppe.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

InvalidDBSecurityGroupState

Der Status der Sicherheitsgruppe lässt das Löschen nicht zu.

HTTP Status Code: 400

InvalidDBSubnetGroupStateFault

Die Subnetzgruppe kann nicht gelöscht werden, da sie verwendet wird.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nicht alle Availability Zones ab, nachdem sie erstellt wurde, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python

- AWS-SDK für Ruby V3

ModifyDBClusterParameterGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändert die Parameter einer Cluster-Parametergruppe. Wenn Sie mehr als einen Parameter ändern möchten, übergeben Sie eine Liste mit den folgenden Optionen: `ParameterName`, `ParameterValue` und `ApplyMethod`. Pro Anforderung können maximal 20 Parameter geändert werden.

> **Note**
>
> Änderungen an dynamischen Parametern werden sofort angewendet. Änderungen an statischen Parametern erfordern einen Neustart oder ein Wartungsfenster, bevor die Änderung wirksam werden kann.

> **Important**
>
> Nachdem Sie eine Cluster-Parametergruppe erstellt haben, sollten Sie mindestens fünf Minuten warten, bevor Sie das erste Cluster erstellen, das diese Cluster-Parametergruppe als Standardparametergruppe verwendet. Auf diese Weise kann Amazon DocumentDB die Erstellungsaktion vollständig abschließen, bevor die Parametergruppe als Standard für einen neuen Cluster verwendet wird. Dieser Schritt ist besonders wichtig für Parameter, die beim Erstellen der Standarddatenbank für einen Cluster von kritischer Bedeutung sind, z. B. den Zeichensatz für die Standarddatenbank, der durch den Parameter `character_set_database` definiert wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der zu ändernden Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

Parameter.Parameter.N

Eine Liste der zu ändernden Parameter in der Cluster-Parametergruppe.

Typ: Array von Parameter-Objekten

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterParameterGroupName

Der Name einer Cluster-Parametergruppe.

Einschränkungen:

- Muss zwischen 1 und 255 Buchstaben oder Zahlen enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

> **Note**
>
> Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

InvalidDBParameterGroupState

Die Parametergruppe wird verwendet oder befindet sich in einem ungültigen Zustand. Wenn Sie versuchen, die Parametergruppe zu löschen, können Sie sie nicht löschen, wenn sich die Parametergruppe in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ModifyDBClusterSnapshotAttribute

Service: Amazon DocumentDB (with MongoDB compatibility)

Fügt einem manuellen Cluster-Snapshot ein Attribut und Werte hinzu oder entfernt ein Attribut und Werte daraus.

Um einen manuellen Cluster-Snapshot für andere freizugebenAWS-Konten, geben Sie `restore` als an und verwenden Sie den Parameter `AttributeName`, `ValuesToAdd` um eine Liste der IDs des hinzuzufügenAWS-Konten, die zur Wiederherstellung des manuellen Cluster-Snapshots autorisiert sind. Verwenden Sie den Wert , `all` um den manuellen Cluster-Snapshot öffentlich zu machen, was bedeutet, dass er von allen kopiert oder wiederhergestellt werden kannAWS-Konten. Fügen Sie nicht den `all` Wert für manuelle Cluster-Snapshots hinzu, die private Informationen enthalten, die nicht für alle verfügbar sein sollenAWS-Konten. Wenn ein manueller Cluster-Snapshot verschlüsselt ist, kann er freigegeben werden, jedoch nur durch Angabe einer Liste autorisierter AWS-Konto IDs für den -`ValuesToAdd`Parameter. Sie können in diesem Fall `all` nicht als Wert für diesen Parameter verwenden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

AttributeName

Der Name des zu ändernden Cluster-Snapshot-Attributs.

Um die Autorisierung für andere AWS-Konten zum Kopieren oder Wiederherstellen eines manuellen Cluster-Snapshots zu verwalten, setzen Sie diesen Wert auf `restore`.

Typ: Zeichenfolge

Erforderlich: Ja

DBClusterSnapshotIdentifier

Die Kennung für den Cluster-Snapshot, für den die Attribute geändert werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

ValuesToAddAttributeValue.N

Eine Liste der Cluster-Snapshot-Attribute, die dem durch angegebenen Attribut hinzugefügt werden sollen`AttributeName`.

Um andere AWS-Konten zum Kopieren oder Wiederherstellen eines manuellen Cluster-Snapshots zu autorisieren, legen Sie diese Liste so fest, dass sie eine oder mehrere AWS-Konto IDs enthält. Um den manuellen Cluster-Snapshot von einem wiederherstellbar zu machenAWS-Konto, setzen Sie ihn auf `all`. Fügen Sie nicht den `all` Wert für manuelle Cluster-Snapshots hinzu, die private Informationen enthalten, die nicht für alle verfügbar sein sollenAWS-Konten.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

ValuesToRemoveAttributeValue.N

Eine Liste der Cluster-Snapshot-Attribute, die aus dem durch angegebenen Attribut entfernt werden sollen`AttributeName`.

Um die Autorisierung für andere AWS-Konten zum Kopieren oder Wiederherstellen eines manuellen Cluster-Snapshots zu entfernen, legen Sie diese Liste so fest, dass sie eine oder mehrere AWS-Konto Kennungen enthält. Um die Autorisierung für ein AWS-Konto zum Kopieren oder Wiederherstellen des Cluster-Snapshots zu entfernen, setzen Sie ihn auf `all` . Wenn Sie angeben`all`, kann ein , AWS-Konto dessen Konto-ID dem `restore` Attribut explizit hinzugefügt wurde, weiterhin einen manuellen Cluster-Snapshot kopieren oder wiederherstellen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterSnapshotAttributesResult

Detaillierte Informationen zu den Attributen, die einem Cluster-Snapshot zugeordnet sind.

Typ: DBClusterSnapshotAttributesResult Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

SharedSnapshotQuotaExceeded

Sie haben die maximale Anzahl an Konten überschritten, für die Sie einen manuellen DB-Snapshot freigeben können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ModifyDBInstance

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändert die Einstellungen für eine Instance. Sie können einen oder mehrere Datenbank-Konfigurationsparameter ändern, indem Sie diese Parameter und die neuen Werte in der Anforderung angeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die Instance-ID. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBInstance` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

ApplyImmediately

Gibt an, ob die Änderungen in dieser Anforderung und alle ausstehenden Änderungen so schnell wie möglich asynchron angewendet werden, unabhängig von der `PreferredMaintenanceWindow` Einstellung für die Instance.

Wenn dieser Parameter auf gesetzt ist`false`, werden Änderungen an der Instance während des nächsten Wartungsfensters übernommen. Einige Parameteränderungen können zu einem Ausfall führen und werden beim nächsten Neustart angewendet.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

AutoMinorVersionUpgrade

Dieser Parameter gilt nicht für Amazon DocumentDB. Amazon DocumentDB führt unabhängig vom eingestellten Wert keine kleineren Versions-Upgrades durch.

Typ: Boolesch

Erforderlich: Nein

CACertificateIdentifier

Gibt das Zertifikat an, das mit der Instance verknüpft werden muss.

Typ: Zeichenfolge

Erforderlich: Nein

CertificateRotationRestart

Gibt an, ob die DB-Instance neu gestartet wird, wenn Sie Ihr SSL/TLS-Zertifikat rotieren.

Standardmäßig wird die DB-Instance neu gestartet, wenn Sie Ihr SSL/TLS-Zertifikat rotieren. Das Zertifikat wird erst aktualisiert, wenn die DB-Instance neu gestartet wurde.

> **Important**
>
> Legen Sie diesen Parameter nur fest, wenn Sie SSL/TLS nicht verwenden, um eine Verbindung mit der DB-Instance herzustellen.

Wenn Sie SSL/TLS verwenden, um eine Verbindung mit der DB-Instance herzustellen, finden Sie weitere Informationen unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten während der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Typ: Boolesch

Erforderlich: Nein

CopyTagsToSnapshot

Ein Wert, der angibt, ob alle Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden sollen. Standardmäßig werden Tags nicht kopiert.

Typ: Boolesch

Erforderlich: Nein

DBInstanceClass

Die neue Rechen- und Speicherkapazität der Instance, z. B. `db.r5.large`. Nicht alle Instance-Klassen sind in allen verfügbarAWS-Regionen.

Wenn Sie die Instance-Klasse ändern, kommt es während der Änderung zu einem Ausfall. Die Änderung wird während des nächsten Wartungsfensters angewendet, es sei denn, `ApplyImmediately` wird für diese Anforderung als `true` angegeben.

Standard: Verwendet vorhandene Einstellungen.

Typ: Zeichenfolge

Erforderlich: Nein

EnablePerformanceInsights

Ein Wert, der angibt, ob Performance Insights für die DB-Instance aktiviert werden sollen. Weitere Informationen finden Sie unter Verwenden von Amazon Performance Insights.

Typ: Boolesch

Erforderlich: Nein

NewDBInstanceIdentifier

Die neue Instance-ID für die Instance beim Umbenennen einer Instance. Wenn Sie die Instance-ID ändern, erfolgt sofort ein Neustart der Instance, wenn Sie `Apply Immediately` auf setzen`true`. Sie tritt während des nächsten Wartungsfensters auf, wenn Sie `Apply Immediately` auf setzen`false`. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `mydbinstance`

Typ: Zeichenfolge

Erforderlich: Nein

PerformanceInsightsKMSKeyId

Die AWS KMS-Schlüssel-ID für die Verschlüsselung von Performance-Insights-Daten.

Die AWS KMS Schlüsselkennung ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel.

Wenn Sie keinen Wert für PerformanceInsightsKMS angebenKeyId, verwendet Amazon DocumentDB Ihren Standard-KMS-Schlüssel. Es gibt einen Standard-KMS-Schlüssel für Ihr Amazon Web Services-Konto. Ihr Amazon-Web-Services-Konto verfügt über einen anderen Standard-KMS-Schlüssel für jede Amazon-Web-Services-Region.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum (in UTC), in dem Systemwartungen durchgeführt werden können, die möglicherweise zu einem Nutzungsausfall führen. Das Ändern dieses Parameters führt nur in der folgenden Situation zu einem Ausfall, und die Änderung wird so schnell wie möglich asynchron angewendet. Wenn ausstehende Aktionen einen Neustart verursachen und das Wartungsfenster so geändert wird, dass es die aktuelle Zeit enthält, führt das Ändern dieses Parameters zu einem Neustart der Instance. Wenn Sie dieses Fenster auf die aktuelle Zeit verschieben, müssen zwischen der aktuellen Zeit und dem Ende des Fensters mindestens 30 Minuten liegen, um sicherzustellen, dass ausstehende Änderungen angewendet werden.

Standard: Verwendet vorhandene Einstellungen.

Format: `ddd:hh24:mi-ddd:hh24:mi`

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: Muss mindestens 30 Minuten betragen.

Typ: Zeichenfolge

Erforderlich: Nein

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance hochgestuft wird.

Standard: 1

Gültige Werte: 0–15

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instance.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AuthorizationNotFound

Die angegebene CIDR-IP oder Amazon EC2-Sicherheitsgruppe ist für die angegebene Sicherheitsgruppe nicht autorisiert.

Amazon DocumentDB ist möglicherweise auch nicht autorisiert, mithilfe von IAM die erforderlichen Aktionen in Ihrem Namen durchzuführen.

HTTP Status Code: 404

CertificateNotFound

`CertificateIdentifier` bezieht sich nicht auf ein vorhandenes Zertifikat.

HTTP Status Code: 404

DBInstanceAlreadyExists

Sie haben bereits eine Instance mit der angegebenen Kennung.

HTTP Status Code: 400

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

DBSecurityGroupNotFound

`DBSecurityGroupName` bezieht sich nicht auf eine vorhandene Sicherheitsgruppe.

HTTP Status Code: 404

DBUpgradeDependencyFailure

Das Upgrade ist fehlgeschlagen, da eine Ressource, von der abhängig ist, nicht geändert werden kann.

HTTP Status Code: 400

InsufficientDBInstanceCapacity

Die angegebene Instance-Klasse ist in der angegebenen Availability Zone nicht verfügbar.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

InvalidDBSecurityGroupState

Der Status der Sicherheitsgruppe lässt das Löschen nicht zu.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nicht alle Availability Zones ab, nachdem sie erstellt wurde, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

StorageTypeNotSupported

Der angegebene Speicher `StorageType` kann nicht mit der DB-Instance verknüpft werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ModifyDBSubnetGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändert eine vorhandene Subnetzgruppe. Subnetzgruppen müssen mindestens ein Subnetz in mindestens zwei Availability Zones in der enthaltenAWS-Region.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupName

Der Name für die Subnetzgruppe. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert. Sie können die Standardsubnetzgruppe nicht ändern.

Einschränkungen: Der Wert muss mit dem Namen einer vorhandenen `DBSubnetGroup` übereinstimmen. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Ja

SubnetIdsSubnetIdentifier.N

Die Amazon EC2-Subnetz-IDs für die Subnetzgruppe.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

DBSubnetGroupDescription

Die Beschreibung für die Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBSubnetGroup

Detaillierte Informationen zu einer Subnetzgruppe.

Typ: DBSubnetGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

DBSubnetQuotaExceededFault

Die Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Subnetzen in einer Subnetzgruppe überschreiten.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

SubnetAlreadyInUse

Das Subnetz wird bereits in der Availability Zone verwendet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ModifyEventSubscription

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändert ein vorhandenes Abonnement für Amazon DocumentDB-Ereignisbenachrichtigungen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SubscriptionName

Der Name des Abonnements für Amazon DocumentDB-Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Erforderlich: Ja

Enabled

Ein boolescher Wert; setzen Sie auf , um das Abonnement `true` zu aktivieren.

Typ: Boolesch

Erforderlich: Nein

EventCategoriesEventCategory.N

Eine Liste von Ereigniskategorien für einen `SourceType`, den Sie abonnieren möchten.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SnsTopicArn

Der Amazon-Ressourcenname (ARN) des SNS-Themas, das für die Ereignisbenachrichtigung erstellt wurde. Der ARN wird von Amazon SNS erstellt, wenn Sie ein Thema erstellen und es abonnieren.

Typ: Zeichenfolge

Erforderlich: Nein

SourceType

Der Typ der Quelle, die die Ereignisse generiert. Wenn Sie beispielsweise über Ereignisse benachrichtigt werden möchten, die von einer Instance generiert werden, setzen Sie diesen Parameter auf `db-instance`. Wenn der Wert nicht angegeben ist, werden alle Ereignisse zurückgegeben.

Zulässige Werte: `db-instance`, `db-parameter-group`, `db-security-group`

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einem Ereignis, das Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

EventSubscriptionQuotaExceeded

Sie haben die maximale Anzahl von Ereignisabonnements erreicht.

HTTP Status Code: 400

SNSInvalidTopic

Amazon SNS hat darauf geantwortet, dass ein Problem mit dem angegebenen Thema vorliegt.

HTTP Status Code: 400

SNSNoAuthorization

Sie sind nicht berechtigt, im Amazon-Ressourcennamen (ARN) des SNS-Themas zu veröffentlichen.

HTTP Status Code: 400

SNSTopicArnNotFound

Der Amazon-Ressourcenname (ARN) des SNS-Themas ist nicht vorhanden.

HTTP Status Code: 404

SubscriptionCategoryNotFound

Die angegebene Kategorie ist nicht vorhanden.

HTTP Status Code: 404

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ModifyGlobalCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändern Sie eine Einstellung für einen globalen Amazon DocumentDB-Cluster. Sie können einen oder mehrere Konfigurationsparameter (z. B. Löschschutz) oder die globale Cluster-ID ändern, indem Sie diese Parameter und die neuen Werte in der Anforderung angeben.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die Kennung für den globalen Cluster, der geändert wird. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der ID eines vorhandenen globalen Clusters übereinstimmen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

DeletionProtection

Gibt an, ob für den globalen Cluster der Löschschutz aktiviert ist. Der globale Cluster kann nicht gelöscht werden, wenn der Löschschutz aktiviert ist.

Typ: Boolesch

Erforderlich: Nein

NewGlobalClusterIdentifier

Die neue Kennung für einen globalen Cluster, wenn Sie einen globalen Cluster ändern. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.

  Das erste Zeichen muss ein Buchstabe sein.

  Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten

Beispiel: `my-cluster2`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

GlobalClusterNotFoundFault

bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, während sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

RebootDBInstance

Service: Amazon DocumentDB (with MongoDB compatibility)

Möglicherweise müssen Sie Ihre Instance neu starten, normalerweise aus Wartungsgründen. Wenn Sie beispielsweise bestimmte Änderungen vornehmen oder die der Instance zugeordnete Cluster-Parametergruppe ändern, müssen Sie die Instance neu starten, damit die Änderungen wirksam werden.

Durch den Neustart einer Instance wird der Datenbank-Engine-Service neu gestartet. Der Neustart einer Instance führt zu einem vorübergehenden Ausfall, bei dem der Instance-Status auf einen Neustart von gesetzt ist.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die Instance-ID. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBInstance` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

ForceFailover

Wenn `true`, wird der Neustart durch ein Multi-AZ-Failover durchgeführt.

Einschränkung: Sie können nicht angeben`true`, wenn die Instance nicht für Multi-AZ konfiguriert ist.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instance.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

RemoveFromGlobalCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Trennt einen sekundären Amazon DocumentDB-Cluster von einem globalen Cluster. Der Cluster wird zu einem eigenständigen Cluster mit Lese-/Schreibfähigkeit, anstatt schreibgeschützt zu sein und Daten von einem primären Cluster in einer anderen Region zu empfangen.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DbClusterIdentifier

Der Amazon-Ressourcenname (ARN), der den Cluster identifiziert, der vom globalen Amazon DocumentDB-Cluster getrennt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

GlobalClusterIdentifier

Die Cluster-ID, die vom globalen Amazon DocumentDB-Cluster getrennt werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

GlobalClusterNotFoundFault

bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, während sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3

- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

RemoveSourceIdentifierFromSubscription

Service: Amazon DocumentDB (with MongoDB compatibility)

Entfernt eine Quell-ID aus einem vorhandenen Abonnement für Amazon DocumentDB-Ereignisbenachrichtigungen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceIdentifier

Die Quell-ID, die aus dem Abonnement entfernt werden soll, z. B. die Instance-ID für eine Instance oder der Name einer Sicherheitsgruppe.

Typ: Zeichenfolge

Erforderlich: Ja

SubscriptionName

Der Name des Abonnements für Amazon DocumentDB-Ereignisbenachrichtigungen, aus dem Sie eine Quell-ID entfernen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einem Ereignis, das Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

SourceNotFound

Die angeforderte Quelle konnte nicht gefunden werden.

HTTP Status Code: 404

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

RemoveTagsFromResource

Service: Amazon DocumentDB (with MongoDB compatibility)

Entfernt Metadaten-Tags aus einer Amazon DocumentDB-Ressource.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ResourceName

Die Amazon DocumentDB-Ressource, aus der die Tags entfernt werden. Dieser Wert ist ein Amazon-Ressourcenname (ARN).

Typ: Zeichenfolge

Erforderlich: Ja

TagKeys.member.N

Der Tag-Schlüssel (Name) des zu entfernenden Tags.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceNotFound

`DBInstanceIdentifier` bezieht sich nicht auf eine vorhandene Instance.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

ResetDBClusterParameterGroup

Service: Amazon DocumentDB (with MongoDB compatibility)

Ändert die Parameter einer Cluster-Parametergruppe auf den Standardwert. Um bestimmte Parameter zurückzusetzen, übermitteln Sie eine Liste der folgenden Optionen: `ParameterName` und `ApplyMethod`. Um die gesamte Cluster-Parametergruppe zurückzusetzen, geben Sie die `ResetAllParameters` Parameter `DBClusterParameterGroupName` und an.

Wenn Sie die gesamte Gruppe zurücksetzen, werden dynamische Parameter sofort aktualisiert und statische Parameter werden auf gesetzt, `pending-reboot` damit sie beim nächsten Neustart der DB-Instance wirksam werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der zurückzusetzenden Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

Parameter.Parameter.N

Eine Liste der Parameternamen in der Cluster-Parametergruppe, die auf die Standardwerte zurückgesetzt werden sollen. Sie können den Parameter nicht verwenden, wenn der Parameter `ResetAllParameters` auf `true` festgelegt ist.

Typ: Array von Parameter-Objekten

Erforderlich: Nein

ResetAllParameters

Ein Wert, der auf gesetzt ist, `true` um alle Parameter in der Cluster-Parametergruppe auf ihre Standardwerte zurückzusetzen, `false` andernfalls . Diesen Parameter können Sie nicht verwenden, wenn es eine Liste von Parameternamen für `Parameters` gibt.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBClusterParameterGroupName

Der Name einer Cluster-Parametergruppe.

Einschränkungen:

- Muss zwischen 1 und 255 Buchstaben oder Zahlen enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

> **Note**
>
> Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName` bezieht sich nicht auf eine vorhandene Parametergruppe.

HTTP Status Code: 404

InvalidDBParameterGroupState

Die Parametergruppe wird verwendet oder befindet sich in einem ungültigen Zustand. Wenn Sie versuchen, die Parametergruppe zu löschen, können Sie sie nicht löschen, wenn sich die Parametergruppe in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

RestoreDBClusterFromSnapshot

Service: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen neuen Cluster aus einem Snapshot oder Cluster-Snapshot.

Wenn ein Snapshot angegeben wird, wird der Ziel-Cluster aus dem Quell-DB-Snapshot mit einer Standardkonfiguration und einer Standardsicherheitsgruppe erstellt.

Wenn ein Cluster-Snapshot angegeben wird, wird der Ziel-Cluster vom Wiederherstellungspunkt des Quell-Clusters aus mit derselben Konfiguration wie der ursprüngliche Quell-DB-Cluster erstellt, mit der Ausnahme, dass der neue Cluster mit der Standardsicherheitsgruppe erstellt wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Der Name des Clusters, der aus dem Snapshot oder dem Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-snapshot-id`

Typ: Zeichenfolge

Erforderlich: Ja

Engine

Die Datenbank-Engine, die für den neuen Cluster verwendet werden soll.

Standard: Gleich wie Quelle.

Einschränkung: Muss mit der Engine der Quelle kompatibel sein.

Typ: Zeichenfolge

Erforderlich: Ja

SnapshotIdentifier

Die Kennung für den Snapshot oder den Cluster-Snapshot, der zur Wiederherstellung verwendet werden soll.

Sie können entweder den Namen oder den Amazon-Ressourcennamen (ARN) verwenden, um einen Cluster-Snapshot festzulegen. Sie können jedoch auch nur den ARN verwenden, um einen Snapshot festzulegen.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen Snapshots übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

AvailabilityZonesAvailabilityZone.N

Stellt die Liste der Amazon EC2 Availability Zones bereit, in denen Instances im wiederhergestellten DB-Cluster erstellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

DBClusterParameterGroupName

Der Name der DB-Cluster-Parametergruppe, die mit diesem DB-Cluster verknüpft werden soll.

Typ: Zeichenfolge. Erforderlich: Nein.

Wenn dieses Argument weggelassen wird, wird die Standard-DB-Cluster-Parametergruppe verwendet. Falls angegeben, muss mit dem Namen einer vorhandenen Standard-DB-Cluster-Parametergruppe übereinstimmen. Die Zeichenfolge muss aus 1 bis 255 Buchstaben, Zahlen oder Bindestrichen bestehen. Das erste Zeichen muss ein Buchstabe sein und darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupName

Der Name der Subnetzgruppe, die für den neuen Cluster verwendet werden soll.

Einschränkungen: Falls angegeben, muss mit dem Namen einer vorhandenen übereinstimmenDBSubnetGroup.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn aktiviert `DeletionProtection` ist, kann der Cluster nur gelöscht werden, wenn er geändert und deaktiviert `DeletionProtection` wird. `DeletionProtection` schützt Cluster vor versehentlichem Löschen.

Typ: Boolesch

Erforderlich: Nein

EnableCloudwatchLogsExports.member.N

Eine Liste der Protokolltypen, die für den Export nach Amazon CloudWatch Logs aktiviert werden müssen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

EngineVersion

Die Version der Datenbank-Engine, die für den neuen Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüsselkennung, die beim Wiederherstellen eines verschlüsselten Clusters aus einem DB-Snapshot oder Cluster-Snapshot verwendet werden soll.

Die Kennung für den AWS KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den AWS KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben wiederherstellenAWS-Konto, dem der AWS KMSVerschlüsselungsschlüssel gehört, der zum Verschlüsseln des neuen Clusters verwendet wurde, können Sie den AWS KMS Schlüsselalias anstelle des ARN für den AWS KMSVerschlüsselungsschlüssel verwenden.

Wenn Sie keinen Wert für den `KmsKeyId`-Parameter angeben, geschieht folgendes:

- Wenn der Snapshot oder der Cluster-Snapshot in verschlüsselt `SnapshotIdentifier` ist, wird der wiederhergestellte Cluster mit dem AWS KMS Schlüssel verschlüsselt, der zum Verschlüsseln des Snapshots oder des Cluster-Snapshots verwendet wurde.
- Wenn der Snapshot oder der Cluster-Snapshot in nicht verschlüsselt `SnapshotIdentifier` ist, wird der wiederhergestellte DB-Cluster nicht verschlüsselt.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Portnummer, an dem der neue Cluster Verbindungen akzeptiert.

Einschränkungen: Muss ein Wert von `1150` bis sein`65535`.

Standard: Derselbe Port wie der ursprüngliche Cluster.

Typ: Ganzzahl

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB-Entwicklerhandbuch.

Gültige Werte für den Speichertyp – `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem wiederhergestellten Cluster zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

VpcSecurityGroupIdsVpcSecurityGroupId.N

Eine Liste der Virtual Private Cloud (VPC)-Sicherheitsgruppen, zu denen der neue Cluster gehören wird.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterQuotaExceededFault

Der Cluster kann nicht erstellt werden, da Sie das maximal zulässige Kontingent für Cluster erreicht haben.

HTTP Status Code: 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

InsufficientDBClusterCapacityFault

Der Cluster verfügt nicht über genügend Kapazität für den aktuellen Vorgang.

HTTP Status Code: 403

InsufficientStorageClusterCapacity

Für die aktuelle Aktion ist nicht genügend Speicherplatz verfügbar. Möglicherweise können Sie diesen Fehler beheben, indem Sie Ihre Subnetzgruppe aktualisieren, um verschiedene Availability Zones zu verwenden, in denen mehr Speicher verfügbar ist.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBSnapshotState

Der Status des Snapshots erlaubt kein Löschen.

HTTP Status Code: 400

InvalidRestoreFault

Sie können keine Wiederherstellung aus einem Virtual Private Cloud (VPC)-Backup auf eine Nicht-VPC-DB-Instance durchführen.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nicht alle Availability Zones ab, nachdem sie erstellt wurde, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen -AWS KMSSchlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

RestoreDBClusterToPointInTime

Service: Amazon DocumentDB (with MongoDB compatibility)

Stellt einen Cluster zu einem beliebigen Zeitpunkt wieder her. Benutzer können den Zustand jedes beliebigen Zeitpunkts vor `LatestRestorableTime` bis zu `BackupRetentionPeriod` Tagen wiederherstellen. Der Ziel-Cluster wird aus dem Quell-Cluster mit derselben Konfiguration wie der ursprüngliche Cluster erstellt, mit der Ausnahme, dass der neue Cluster mit der Standardsicherheitsgruppe erstellt wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Der Name des neuen Clusters, der erstellt werden soll.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Ja

SourceDBClusterIdentifier

Die Kennung des Quell-Clusters, von dem wiederhergestellt werden soll.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

DBSubnetGroupName

Der Subnetzgruppenname, der für den neuen Cluster verwendet werden soll.

Einschränkungen: Falls angegeben, muss mit dem Namen einer vorhandenen übereinstimmenDBSubnetGroup.

Beispiel: mySubnetgroup

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn aktiviert DeletionProtection ist, kann der Cluster nur gelöscht werden, wenn er geändert und deaktiviert DeletionProtection wird. DeletionProtection schützt Cluster vor versehentlichem Löschen.

Typ: Boolesch

Erforderlich: Nein

EnableCloudwatchLogsExports.member.N

Eine Liste der Protokolltypen, die für den Export nach Amazon CloudWatch Logs aktiviert werden müssen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüsselkennung, die bei der Wiederherstellung eines verschlüsselten Clusters aus einem verschlüsselten Cluster verwendet werden soll.

Die Kennung für den AWS KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den AWS KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben wiederherstellenAWS-Konto, dem der AWS KMSVerschlüsselungsschlüssel gehört, der zum Verschlüsseln des neuen Clusters verwendet wurde, können Sie den AWS KMS Schlüsselalias anstelle des ARN für den AWS KMSVerschlüsselungsschlüssel verwenden.

Sie können in einem neuen Cluster wiederherstellen und den neuen Cluster mit einem -AWS KMSSchlüssel verschlüsseln, der sich von dem AWS KMS Schlüssel unterscheidet, der zur Verschlüsselung des Quell-Clusters verwendet wurde. Der neue DB-Cluster wird mit dem Schlüssel verschlüsseltAWS KMS, der durch den -KmsKeyIdParameter identifiziert wird.

Wenn Sie keinen Wert für den `KmsKeyId`-Parameter angeben, geschieht folgendes:

- Wenn der Cluster verschlüsselt ist, wird der wiederhergestellte Cluster mit dem -AWS KMSSchlüssel verschlüsselt, der zur Verschlüsselung des Quell-Clusters verwendet wurde.
- Wenn der Cluster nicht verschlüsselt ist, wird der wiederhergestellte Cluster nicht verschlüsselt.

Wenn sich auf einen Cluster `DBClusterIdentifier` bezieht, der nicht verschlüsselt ist, wird die Wiederherstellungsanforderung abgelehnt.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Portnummer, an dem der neue Cluster Verbindungen akzeptiert.

Einschränkungen: Muss ein Wert von `1150` bis sein`65535`.

Standard: Der Standardport für die Engine.

Typ: Ganzzahl

Erforderlich: Nein

RestoreToTime

Das Datum und die Uhrzeit für die Wiederherstellung des Clusters.

Gültige Werte: Eine Uhrzeit im UTC-Format (Universal Coordinated Time).

Einschränkungen:

- Muss vor dem letzten wiederherstellbaren Zeitpunkt für die Instance liegen.
- Muss angegeben werden, wenn der Parameter `UseLatestRestorableTime` nicht angegeben ist.
- Kann nicht angegeben werden, wenn der Parameter `UseLatestRestorableTime` auf `true` festgelegt ist.
- Kann nicht angegeben werden, wenn der Parameter `RestoreType` auf `copy-on-write` festgelegt ist.

Beispiel: `2015-03-07T23:45:00Z`

Typ: Zeitstempel

Erforderlich: Nein

RestoreType

Der Typ der auszuführenden Wiederherstellung. Sie können einen der folgenden Werte angeben:

- `full-copy` – Der neue DB-Cluster wird als vollständige Kopie des Quell-DB-Clusters wiederhergestellt.
- `copy-on-write` – Der neue DB-Cluster wird als Klon des Quell-DB-Clusters wiederhergestellt.

Einschränkungen: Sie können `copy-on-write` nicht angeben, wenn die Engine-Version des Quell-DB-Clusters älter als 1.11 ist.

Wenn Sie keinen Wert für `RestoreType` angeben, wird der neue DB-Cluster als vollständige Kopie des Quell-DB-Clusters wiederhergestellt.

Typ: Zeichenfolge

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB-Entwicklerhandbuch.

Gültige Werte für den Speichertyp – `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem wiederhergestellten Cluster zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

UseLatestRestorableTime

Ein Wert, der auf `true` festgelegt ist, um den Cluster auf den letzten wiederherstellbaren Sicherungszeitpunkt wiederherzustellen, und sonst `false` anzeigt.

Standard: `false`

Einschränkungen: Darf nicht angegeben werden, wenn der Parameter `RestoreToTime` angegeben ist.

Typ: Boolesch

Erforderlich: Nein

VpcSecurityGroupIdsVpcSecurityGroupId.N

Eine Liste der VPC-Sicherheitsgruppen, zu denen der neue Cluster gehört.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterQuotaExceededFault

Der Cluster kann nicht erstellt werden, da Sie das maximal zulässige Kontingent für Cluster erreicht haben.

HTTP Status Code: 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier` bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName` bezieht sich nicht auf eine vorhandene Subnetzgruppe.

HTTP Status Code: 404

InsufficientDBClusterCapacityFault

Der Cluster verfügt nicht über genügend Kapazität für den aktuellen Vorgang.

HTTP Status Code: 403

InsufficientStorageClusterCapacity

Für die aktuelle Aktion ist nicht genügend Speicherplatz verfügbar. Möglicherweise können Sie diesen Fehler beheben, indem Sie Ihre Subnetzgruppe aktualisieren, um verschiedene Availability Zones zu verwenden, in denen mehr Speicher verfügbar ist.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBSnapshotState

Der Status des Snapshots erlaubt kein Löschen.

HTTP Status Code: 400

InvalidRestoreFault

Sie können keine Wiederherstellung aus einem Virtual Private Cloud (VPC)-Backup auf eine Nicht-VPC-DB-Instance durchführen.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist nicht gültig oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nicht alle Availability Zones ab, nachdem sie erstellt wurde, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen -AWS KMSSchlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anforderung würde dazu führen, dass Sie die zulässige Speichermenge überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

StartDBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Startet den gestoppten Cluster, der durch angegeben wird, neu`DBClusterIdentifier`. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Kennung des Clusters, der neu gestartet werden soll. Beispiel: `docdb-2019-05-28-15-24-52`

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

StopDBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Stoppt den laufenden Cluster, der durch angegeben wird`DBClusterIdentifier`. Der Cluster muss sich im Status „Verfügbar“ befinden. Weitere Informationen finden Sie unter Stoppen und Starten eines Amazon DocumentDB-Clusters.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Kennung des Clusters, der angehalten werden soll. Beispiel: `docdb-2019-05-28-15-24-52`

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier` bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instance befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-Befehlszeilenschnittstelle
- AWS-SDK für .NET
- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für JavaScript V3
- AWS-SDK für PHP V3
- AWS-SDK für Python
- AWS-SDK für Ruby V3

Amazon DocumentDB Elastic Clusters

Die folgenden Aktionen werden von Amazon DocumentDB Elastic Clustern unterstützt:

- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster

- GetClusterSnapshot
- ListClusters
- ListClusterSnapshots
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster
- TagResource
- UntagResource
- UpdateCluster

CopyClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Kopiert einen Snapshot eines elastischen Clusters.

Anforderungssyntax

```
POST /cluster-snapshot/snapshotArn/copy HTTP/1.1
Content-type: application/json

{
   "copyTags": boolean,
   "kmsKeyId": "string",
   "tags": {
      "string" : "string"
   },
   "targetSnapshotName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die Amazon-Ressourcenname (ARN)-ID des elastischen Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

targetSnapshotName

Die Kennung des neuen elastischen Cluster-Snapshots, der aus dem Quell-Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.

- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `elastic-cluster-snapshot-5`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 63 Zeichen.

Erforderlich: Ja

copyTags

Legen Sie den Wert auf fest`true`, um alle Tags aus dem Quell-Cluster-Snapshot in den elastischen Ziel-Cluster-Snapshot zu kopieren. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

kmsKeyId

Die AWS KMS-Schlüssel-ID für einen verschlüsselten elastischen Cluster-Snapshot. Die AWS KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN), die AWS KMS-Schlüsselkennung oder der AWS KMS-Schlüsselalias für den AWS KMS-Verschlüsselungsschlüssel.

Wenn Sie einen verschlüsselten Snapshot eines elastischen Clusters aus Ihrem AWS Konto kopieren, können Sie einen Wert für angeben, `KmsKeyId` um die Kopie mit einem neuen AWSS KMS-Verschlüsselungsschlüssel zu verschlüsseln. Wenn Sie keinen Wert für angeben`KmsKeyId`, wird die Kopie des elastischen Cluster-Snapshots mit demselben `AWS` KMS-Schlüssel wie der elastische Quell-Cluster-Snapshot verschlüsselt.

Um einen verschlüsselten Snapshot eines elastischen Clusters in eine andere AWS Region zu kopieren, setzen Sie `KmsKeyId` auf die AWS KMS-Schlüssel-ID, die Sie zum Verschlüsseln der Kopie des Snapshots des elastischen Clusters in der Zielregion verwenden möchten. AWS KMS-Verschlüsselungsschlüssel sind spezifisch für die AWS Region, in der sie erstellt werden, und Sie können keine Verschlüsselungsschlüssel aus einer AWS Region in einer anderen AWS Region verwenden.

Wenn Sie einen unverschlüsselten Snapshot eines elastischen Clusters kopieren und einen Wert für den `KmsKeyId` Parameter angeben, wird ein Fehler zurückgegeben.

Typ: Zeichenfolge

Erforderlich: Nein

tags

Die Tags, die dem elastischen Cluster-Snapshot zugewiesen werden sollen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen zu einem bestimmten Elastic-Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ServiceQuotaExceededException

Das Servicekontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

CreateCluster

Service: Amazon DocumentDB Elastic Clusters

Erstellt einen neuen elastischen Amazon DocumentDB-Cluster und gibt dessen Clusterstruktur zurück.

Anforderungssyntax

```
POST /cluster HTTP/1.1
Content-type: application/json

{
   "adminUserName": "string",
   "adminUserPassword": "string",
   "authType": "string",
   "backupRetentionPeriod": number,
   "clientToken": "string",
   "clusterName": "string",
   "kmsKeyId": "string",
   "preferredBackupWindow": "string",
   "preferredMaintenanceWindow": "string",
   "shardCapacity": number,
   "shardCount": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "tags": {
      "string" : "string"
   },
   "vpcSecurityGroupIds": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

adminUserName

Der Name des Administrators für elastische Amazon DocumentDB-Cluster.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben oder Zahlen enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Dies darf kein reserviertes Wort sein.

Typ: Zeichenfolge

Erforderlich: Ja

adminUserPassword

Das Passwort für den Administrator für elastische Amazon DocumentDB-Cluster. Das Passwort kann alle druckbaren ASCII-Zeichen enthalten.

Einschränkungen:

- Muss zwischen 8 und 100 Zeichen enthalten.
- Darf keinen Schrägstrich (/), doppelten Anführungszeichen (") oder das At-Symbol (@) enthalten.

Typ: Zeichenfolge

Erforderlich: Ja

authType

Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das Passwort abgerufen werden soll, das für den Zugriff auf den elastischen Cluster verwendet wird. Gültige Typen sind `PLAIN_TEXT` oder `SECRET_ARN`.

Typ: Zeichenfolge

Zulässige Werte: `PLAIN_TEXT | SECRET_ARN`

Erforderlich: Ja

clusterName

Der Name des neuen elastischen Clusters. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.

- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster`

Typ: Zeichenfolge

Erforderlich: Ja

shardCapacity

Die Anzahl der vCPUs, die jedem elastischen Cluster-Shard zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.

Typ: Ganzzahl

Erforderlich: Ja

shardCount

Die Anzahl der dem elastischen Cluster zugewiesenen Shards. Das Maximum ist 32.

Typ: Ganzzahl

Erforderlich: Ja

backupRetentionPeriod

Die Anzahl der Tage, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

clientToken

Das Client-Token für den elastischen Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

kmsKeyId

Die KMS-Schlüsselkennung, die zum Verschlüsseln des neuen elastischen Clusters verwendet werden soll.

Die Kennung für den KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon-Konto erstellen, das diesen KMS-Verschlüsselungsschlüssel besitzt, können Sie den KMS-Schlüsselalias anstelle des ARN als KMS-Verschlüsselungsschlüssel verwenden.

Wenn kein Verschlüsselungsschlüssel angegeben ist, verwendet Amazon DocumentDB den Standardverschlüsselungsschlüssel, den KMS für Ihr Konto erstellt. Ihr Konto verfügt für jede Amazon-Region über einen anderen Standardverschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

preferredBackupWindow

Der tägliche Zeitraum, in dem automatisierte Backups erstellt werden, wenn automatisierte Backups aktiviert sind, wie durch die festgelegt`backupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

preferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Standard: ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock für jede ausgewählt wird AWS-Regionund an einem zufälligen Wochentag stattfindet.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikat-Instances, die für alle Shards im elastischen Cluster gelten. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt und alle

zusätzlichen Instances Replikate sind, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

subnetIds

Die Amazon EC2-Subnetz-IDs für den neuen elastischen Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

tags

Die Tags, die dem neuen elastischen Cluster zugewiesen werden sollen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

vpcSecurityGroupIds

Eine Liste der EC2-VPC-Sicherheitsgruppen, die dem neuen elastischen Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Der neue Elastic Cluster, der erstellt wurde.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ServiceQuotaExceededException

Das Servicekontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

CreateClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Erstellt einen Snapshot eines elastischen Clusters.

Anforderungssyntax

```
POST /cluster-snapshot HTTP/1.1
Content-type: application/json

{
   "clusterArn": "string",
   "snapshotName": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

clusterArn

Die ARN-ID des elastischen Clusters, von dem Sie einen Snapshot erstellen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotName

Der Name des neuen elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 63 Zeichen.

Erforderlich: Ja

tags

Die Tags, die dem neuen elastischen Cluster-Snapshot zugewiesen werden sollen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen über den neuen Snapshot des elastischen Clusters zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ServiceQuotaExceededException

Das Servicekontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

DeleteCluster

Service: Amazon DocumentDB Elastic Clusters

Löschen Sie einen elastischen Cluster.

Anforderungssyntax

```
DELETE /cluster/clusterArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters, der gelöscht werden soll.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über den neu gelöschten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

DeleteClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Löschen Sie einen elastischen Cluster-Snapshot.

Anforderungssyntax

```
DELETE /cluster-snapshot/snapshotArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die ARN-ID des zu löschenden elastischen Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
```

```
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen über den neu gelöschten elastischen Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

GetCluster

Service: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu einem bestimmten elastischen Cluster zurück.

Anforderungssyntax

```
GET /cluster/clusterArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen zu einem bestimmten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

GetClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu einem bestimmten Elastic-Cluster-Snapshot zurück

Anforderungssyntax

```
GET /cluster-snapshot/snapshotArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die ARN-ID des elastischen Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
```

```
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen zu einem bestimmten Elastic-Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

ListClusters

Service: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu bereitgestellten elastischen Amazon DocumentDB-Clustern zurück.

Anforderungssyntax

```
GET /clusters?maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

maxResults

Die maximale Anzahl der elastischen Cluster-Snapshots, die in der Antwort empfangen werden sollen.

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

nextToken

Ein Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wurde. Wenn dieser Parameter angegeben ist, enthält die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von angegebenen Wert`max-results`.

Wenn es in der einmalige keine Daten mehr gibt, `nextToken` wird die nicht zurückgegeben.

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "clusters": [
      {
         "clusterArn": "string",
         "clusterName": "string",
```

```
            "status": "string"
        }
    ],
    "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

clusters

Eine Liste der elastischen Amazon DocumentDB-Cluster.

Typ: Array von ClusterInList-Objekten

nextToken

Ein Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wurde. Wenn dieser Parameter angegeben ist, enthält die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von angegebenen Wert`max-results`.

Wenn es in der einmalige keine Daten mehr gibt, `nextToken` wird die nicht zurückgegeben.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

ListClusterSnapshots

Service: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu Snapshots für einen angegebenen elastischen Cluster zurück.

Anforderungssyntax

```
GET /cluster-snapshots?
clusterArn=clusterArn&maxResults=maxResults&nextToken=nextToken&snapshotType=snapshotType
 HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

maxResults

Die maximale Anzahl der elastischen Cluster-Snapshots, die in der Antwort empfangen werden sollen.

Gültiger Bereich: Mindestwert von 20. Maximalwert 100.

nextToken

Ein Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wurde. Wenn dieser Parameter angegeben ist, enthält die Antwort nur Datensätze über dieses Token hinaus, bis zu dem von angegebenen Wert`max-results`.

Wenn es keine Daten mehr in der Bolonce gibt, `nextToken` wird die nicht zurückgegeben.

snapshotType

Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:

- `automated` – Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr AWS Konto erstellt hat.
- `manual` – Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr AWS Konto erstellt haben.

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "snapshots": [
      {
         "clusterArn": "string",
         "snapshotArn": "string",
         "snapshotCreationTime": "string",
         "snapshotName": "string",
         "status": "string"
      }
   ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

nextToken

Ein Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wurde. Wenn dieser Parameter angegeben ist, enthält die Antwort nur Datensätze über dieses Token hinaus, bis zu dem von angegebenen Wert`max-results`.

Wenn es keine Daten mehr in der Bolonce gibt, `nextToken` wird die nicht zurückgegeben.

Typ: Zeichenfolge

snapshots

Eine Liste von Snapshots für einen angegebenen elastischen Cluster.

Typ: Array von ClusterSnapshotInList-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3

- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

ListTagsForResource

Service: Amazon DocumentDB Elastic Clusters

Listet alle Tags auf einer Elastic-Cluster-Ressource auf

Anforderungssyntax

```
GET /tags/resourceArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Die ARN-ID der Elastic-Cluster-Ressource.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge von 1011.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

tags

Die Liste der Tags für die angegebene Elastic-Cluster-Ressource.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

RestoreClusterFromSnapshot

Service: Amazon DocumentDB Elastic Clusters

Stellt einen elastischen Cluster aus einem Snapshot wieder her.

Anforderungssyntax

```
POST /cluster-snapshot/snapshotArn/restore HTTP/1.1
Content-type: application/json

{
   "clusterName": "string",
   "kmsKeyId": "string",
   "shardCapacity": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "tags": {
      "string" : "string"
   },
   "vpcSecurityGroupIds": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die ARN-ID des elastischen Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

clusterName

Der Name des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

kmsKeyId

Die KMS-Schlüsselkennung, die zum Verschlüsseln des neuen elastischen Cluster-Clusters von Amazon DocumentDB verwendet werden soll.

Die Kennung für den KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon-Konto erstellen, das diesen KMS-Verschlüsselungsschlüssel besitzt, können Sie den KMS-Schlüsselalias anstelle des ARN als KMS-Verschlüsselungsschlüssel verwenden.

Wenn hier kein Verschlüsselungsschlüssel angegeben ist, verwendet Amazon DocumentDB den Standardverschlüsselungsschlüssel, den KMS für Ihr Konto erstellt. Ihr Konto verfügt für jede Amazon-Region über einen anderen Standardverschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

shardCapacity

Die Kapazität jedes Shards im neuen wiederhergestellten elastischen Cluster.

Typ: Ganzzahl

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikat-Instances, die auf alle Shards im elastischen Cluster angewendet werden. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt und alle zusätzlichen Instances Replikate sind, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

subnetIds

Die Amazon EC2-Subnetz-IDs für den elastischen Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

tags

Eine Liste der Tag-Namen, die dem wiederhergestellten elastischen Cluster zugewiesen werden sollen, in Form eines Arrays von Schlüssel-Wert-Paaren, in dem der Schlüssel der Tag-Name und der Wert der Schlüsselwert ist.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

vpcSecurityGroupIds

Eine Liste der EC2-VPC-Sicherheitsgruppen, die dem elastischen Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
```

```
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen zu einem wiederhergestellten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ServiceQuotaExceededException

Das Servicekontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3

- AWS SDK für Python
- AWS SDK für Ruby V3

StartCluster

Service: Amazon DocumentDB Elastic Clusters

Startet den angehaltenen elastischen Cluster neu, der durch angegeben wird`clusterARN`.

Anforderungssyntax

```
POST /cluster/clusterArn/start HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen zu einem bestimmten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

StopCluster

Service: Amazon DocumentDB Elastic Clusters

Stoppt den laufenden elastischen Cluster, der durch angegeben wird`clusterArn`. Der elastische Cluster muss sich im Status „Verfügbar“ befinden.

Anforderungssyntax

```
POST /cluster/clusterArn/stop HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
```

```
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen zu einem bestimmten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

TagResource

Service: Amazon DocumentDB Elastic Clusters

Fügt Metadaten-Tags zu einer Elastic-Cluster-Ressource hinzu

Anforderungssyntax

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Die ARN-ID der Elastic-Cluster-Ressource.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge von 1011.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

tags

Die Tags, die der Elastic-Cluster-Ressource zugewiesen sind.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Einschränkungen der Wertlänge: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

UntagResource

Service: Amazon DocumentDB Elastic Clusters

Entfernt Metadaten-Tags aus einer Elastic-Cluster-Ressource

Anforderungssyntax

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Die ARN-ID der Elastic-Cluster-Ressource.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge von 1011.

Erforderlich: Ja

tagKeys

Die Tag-Schlüssel, die aus der Elastic-Cluster-Ressource entfernt werden sollen.

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 50 Elemente.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++

- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

UpdateCluster

Service: Amazon DocumentDB Elastic Clusters

Ändert einen elastischen Cluster. Dazu gehören das Aktualisieren von admin-username/password, das Aktualisieren der API-Version und das Einrichten eines Sicherungsfensters und eines Wartungsfensters

Anforderungssyntax

```
PUT /cluster/clusterArn HTTP/1.1
Content-type: application/json

{
   "adminUserPassword": "string",
   "authType": "string",
   "backupRetentionPeriod": number,
   "clientToken": "string",
   "preferredBackupWindow": "string",
   "preferredMaintenanceWindow": "string",
   "shardCapacity": number,
   "shardCount": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "vpcSecurityGroupIds": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

adminUserPassword

Das Passwort, das dem Elastic-Cluster-Administrator zugeordnet ist. Dieses Passwort kann alle druckbaren ASCII-Zeichen, außer Schrägstrich (/), doppeltes Anführungszeichen (") oder das "At"-Zeichen (@), enthalten.

Einschränkungen: Muss zwischen 8 und 100 Zeichen enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

authType

Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das Passwort abgerufen werden soll, das für den Zugriff auf den elastischen Cluster verwendet wird. Gültige Typen sind `PLAIN_TEXT` oder `SECRET_ARN`.

Typ: Zeichenfolge

Zulässige Werte: `PLAIN_TEXT | SECRET_ARN`

Erforderlich: Nein

backupRetentionPeriod

Die Anzahl der Tage, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

clientToken

Das Client-Token für den elastischen Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

preferredBackupWindow

Der tägliche Zeitraum, in dem automatisierte Backups erstellt werden, wenn automatisierte Backups aktiviert sind, wie durch die festgelegt`backupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

preferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Standard: ein 30-minütiges Fenster, das zufällig aus einem 8-Stunden-Zeitblock für jedes ausgewählt wird AWS-Regionund an einem zufälligen Wochentag stattfindet.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

shardCapacity

Die Anzahl der vCPUs, die jedem elastischen Cluster-Shard zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.

Typ: Ganzzahl

Erforderlich: Nein

shardCount

Die Anzahl der dem elastischen Cluster zugewiesenen Shards. Das Maximum ist 32.

Typ: Ganzzahl

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikat-Instances, die auf alle Shards im elastischen Cluster angewendet werden. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt und alle zusätzlichen Instances Replikate sind, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

subnetIds

Die Amazon EC2-Subnetz-IDs für den elastischen Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

vpcSecurityGroupIds

Eine Liste der EC2-VPC-Sicherheitsgruppen, die dem elastischen Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
```

```
            }
        ],
        "status": "string",
        "subnetIds": [ "string" ],
        "vpcSecurityGroupIds": [ "string" ]
    }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über den aktualisierten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es ist ein Zugriffskonflikt aufgetreten.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anforderung aufgrund der Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS -Befehlszeilenschnittstelle
- AWS SDK für .NET
- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für JavaScript V3
- AWS SDK für PHP V3
- AWS SDK für Python
- AWS SDK für Ruby V3

Datentypen

Die folgenden Datentypen werden von unterstützt Amazon DocumentDB (with MongoDB compatibility):

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet

- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

Die folgenden Datentypen werden von Amazon DocumentDB Elastic Clustern unterstützt:

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- Shard
- ValidationExceptionField

Amazon DocumentDB (with MongoDB compatibility)

Die folgenden Datentypen werden unterstützt vonAmazon DocumentDB (with MongoDB compatibility):

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo

- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

AvailabilityZone

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Informationen zu einer Availability Zone.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Name

Der Name der Availability Zone.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Certificate

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein das Datum für einAWS-Konto.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

CertificateArn

Der Amazon-Ressourcenname (ARN) für das Zertifikat.

Beispiel: `arn:aws:rds:us-east-1::cert:rds-ca-2019`

Typ: Zeichenfolge

Required: No

CertificateIdentifier

Der eindeutige Schlüssel, der ein das Zertifikat identifiziert.

Beispiel: `rds-ca-2019`

Typ: Zeichenfolge

Required: No

CertificateType

Der Typ des Zertifikats.

Beispiel: `CA`

Typ: Zeichenfolge

Required: No

Thumbprint

Der Fingerabdruck des Zertifikats.

Typ: Zeichenfolge

Required: No

ValidFrom

Das Datum, nach dem das Zertifikat gültig ist.

Beispiel: `2019-07-31T17:57:09Z`

Typ: Zeitstempel

Required: No

ValidTill

Das Datum, nach dem das Zertifikat nicht mehr gültig ist.

Beispiel: `2024-07-31T17:57:09Z`

Typ: Zeitstempel

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

CertificateDetails

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt die Details des Serverzertifikats der DB-Instance zurück.

Weitere Informationen finden Sie unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten bei der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

CAIdentifier

Die CA-ID des CA-Zertifikats, das für das Serverzertifikat der DB-Instance verwendet wird.

Typ: Zeichenfolge

Required: No

ValidTill

Das Ablaufdatum des Serverzertifikats der DB-Instance.

Typ: Zeitstempel

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

CloudwatchLogsExportConfiguration

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Konfigurationseinstellung für die Protokolltypen, die für den Export in Amazon CloudWatch Logs für eine für eine bestimmte Instance oder für eine bestimmte Instance oder einen bestimmten Cluster aktiviert werden sollen.

Die`EnableLogTypes``DisableLogTypes` And-Arrays bestimmen, welche Logs in CloudWatch Logs exportiert (oder nicht exportiert) werden. Die Werte in diesen Arrays hängen von der verwendeten Engine ab.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DisableLogTypes.member.N

Die Liste der Protokolltypen, die deaktiviert werden sollen.

Typ: Zeichenfolge-Array

Required: No

EnableLogTypes.member.N

Die Liste der Protokolltypen, die aktiviert werden sollen.

Typ: Zeichenfolge-Array

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go

- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBCluster

Service: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem Cluster.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AssociatedRoles.DBClusterRole.N

Stellt eine Liste der AWS Identity and Access Management (IAM)-Rollen bereit, die dem Cluster zugeordnet sind. (IAM)-Rollen, die einem Cluster zugeordnet sind, erteilen dem Cluster die Berechtigung, in Ihrem Namen auf andere -AWSServices zuzugreifen.

Typ: Array von DBClusterRole-Objekten

Erforderlich: Nein

AvailabilityZones.AvailabilityZone.N

Stellt die Liste der Amazon EC2 Availability Zones bereit, in denen Instances im Cluster erstellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

BackupRetentionPeriod

Gibt die Anzahl der Tage an, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

CloneGroupId

Identifiziert die Clone-Gruppe, mit der der DB-Cluster verknüpft ist.

Typ: Zeichenfolge

Erforderlich: Nein

ClusterCreateTime

Gibt den Zeitpunkt an, zu dem der Cluster erstellt wurde, in UTC (Universal Coordinated Time).

Typ: Zeitstempel

Erforderlich: Nein

DBClusterArn

Der Amazon-Ressourcenname (ARN) für den Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterIdentifier

Enthält eine vom Benutzer bereitgestellte Cluster-ID. Diese Kennung ist der eindeutige Schlüssel, der einen Cluster identifiziert.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterMembers.DBClusterMember.N

Stellt die Liste der Instances bereit, aus denen der Cluster besteht.

Typ: Array von DBClusterMember-Objekten

Erforderlich: Nein

DBClusterParameterGroup

Gibt den Namen der Cluster-Parametergruppe für den Cluster an.

Typ: Zeichenfolge

Erforderlich: Nein

DbClusterResourceId

Die AWS-Regioneindeutige, unveränderliche Kennung für den Cluster. Diese Kennung wird in AWS CloudTrail Protokolleinträgen gefunden, wenn auf den AWS KMS Schlüssel für den Cluster zugegriffen wird.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroup

Gibt Informationen zur Subnetzgruppe an, die dem Cluster zugeordnet ist, einschließlich Name, Beschreibung und Subnetze in der Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn aktiviert `DeletionProtection` ist, kann der Cluster nur gelöscht werden, wenn er geändert und deaktiviert `DeletionProtection` wird. `DeletionProtection` schützt Cluster vor versehentlichem Löschen.

Typ: Boolesch

Erforderlich: Nein

EarliestRestorableTime

Der früheste Zeitpunkt, zu dem eine Datenbank mit point-in-time einer Wiederherstellung wiederhergestellt werden kann.

Typ: Zeitstempel

Erforderlich: Nein

EnabledCloudwatchLogsExports.member.N

Eine Liste von Protokolltypen, die dieser Cluster für den Export nach Amazon CloudWatch Logs konfiguriert hat.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Endpoint

Gibt den Verbindungsendpunkt für die primäre Instance des Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Gibt den Namen der Datenbank-Engine an, die für diesen Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Erforderlich: Nein

HostedZoneId

Gibt die ID an, die Amazon Route 53 zuweist, wenn Sie eine gehostete Zone erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

KmsKeyId

Wenn `StorageEncrypted` ist`true`, die AWS KMS Schlüsselkennung für den verschlüsselten Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

LatestRestorableTime

Gibt den letzten Zeitpunkt an, zu dem eine Datenbank mit point-in-time einer Wiederherstellung wiederhergestellt werden kann.

Typ: Zeitstempel

Erforderlich: Nein

MasterUsername

Enthält den Hauptbenutzernamen für den Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

MultiAZ

Gibt an, ob der Cluster Instances in mehreren Availability Zones hat.

Typ: Boolesch

Erforderlich: Nein

PercentProgress

Gibt den Fortschritt der Operation als Prozentsatz an.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Gibt die Portnummer an, die von der Datenbank-Engine überwacht wird.

Typ: Ganzzahl

Erforderlich: Nein

PreferredBackupWindow

Gibt den täglichen Zeitraum in koordinierter Weltzeit (UTC) an, in dem automatische Sicherungen erstellt werden, wenn automatische Sicherungen aktiviert sind, gemäß `BackupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Gibt den wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time) an.

Typ: Zeichenfolge

Erforderlich: Nein

ReaderEndpoint

Der Leser-Endpunkt für den Cluster. Der Reader-Endpunkt für einen Cluster gleicht die Verbindungen zwischen den Amazon DocumentDB-Replikaten aus, die in einem Cluster verfügbar sind. Wenn Clients neue Verbindungen zum Reader-Endpunkt anfordern, verteilt Amazon DocumentDB die Verbindungsanfragen auf die Amazon DocumentDB-Replikate im Cluster. Diese Funktionalität kann dazu beitragen, Ihren Lese-Workload auf mehrere Amazon DocumentDB-Replikate in Ihrem Cluster auszugleichen.

Wenn ein Failover auftritt und das Amazon DocumentDB-Replikat, mit dem Sie verbunden sind, zur primären Instance hochgestuft wird, wird Ihre Verbindung getrennt. Um Ihren Lese-Workload weiterhin an andere Amazon DocumentDB-Replikate im Cluster zu senden, können Sie sich dann wieder mit dem Reader-Endpunkt verbinden.

Typ: Zeichenfolge

Erforderlich: Nein

ReadReplicaIdentifiers.ReadReplicaIdentifier.N

Enthält eine oder mehrere Kennungen der sekundären Cluster, die diesem Cluster zugeordnet sind.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

ReplicationSourceIdentifier

Enthält die Kennung des Quell-Clusters, wenn es sich bei diesem Cluster um einen sekundären Cluster handelt.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Gibt den aktuellen Status dieses Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Gibt an, ob der Cluster verschlüsselt ist.

Typ: Boolesch

Erforderlich: Nein

StorageType

Speichertyp, der Ihrem Cluster zugeordnet ist

Speichertyp, der Ihrem Cluster zugeordnet ist

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB-Entwicklerhandbuch.

Gültige Werte für den Speichertyp – `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

VpcSecurityGroups.VpcSecurityGroupMembership.N

Stellt eine Liste der Virtual Private Cloud (VPC)-Sicherheitsgruppen bereit, zu denen der Cluster gehört.

Typ: Array von VpcSecurityGroupMembership-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS-SDK für Ruby V3

DBClusterMember

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält Informationen über eine Instanz, die Teil eines Clusters ist.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterParameterGroupStatus

Gibt Sie den Status der Cluster-Parametergruppe für Ihren die Cluster-Parametergruppe für Ihren.

Typ: Zeichenfolge

Required: No

DBInstanceIdentifier

Gibt die Instanz-ID für dieses Mitglied des Clusters an.

Typ: Zeichenfolge

Required: No

IsClusterWriter

Ein Wert, der angibt,`true` ob das Clustermitglied die primäre Instanz für den Cluster ist und`false` andernfalls.

Typ: Boolesch

Required: No

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der Amazon DocumentDB Aurora-Replikat nach einem Ausfall der bestehenden primären Instance zur primären Instance hochgestuft wird.

Typ: Ganzzahl

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Cluster-Parametergruppe.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterParameterGroupArn

Der Amazon-Parametergruppe (ARN) für die Cluster-Parametergruppe.

Typ: Zeichenfolge

Required: No

DBClusterParameterGroupName

Gibt den Name der Cluster-Parametergruppe an.

Typ: Zeichenfolge

Required: No

DBParameterGroupFamily

Gibt den Namen der Parametergruppen-Familie an, mit der diese Cluster-Parametergruppe kompatibel ist.

Typ: Zeichenfolge

Required: No

Description

Stellt die vom Kunden angegebene Beschreibung für diese Cluster-Parametergruppe bereit.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBClusterRole

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Beschreibt eineAWS Identity and Access Management (IAM) -Rolle, die dem Cluster zugeordnet ist.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

RoleArn

Der Amazon-Ressourcenname (ARN) der IAMRolle, die dem DB-Cluster zugeordnet ist.

Typ: Zeichenfolge

Required: No

Status

Beschreibt den Status der Assoziation zwischen der IamRole und dem Cluster. Die`Status` Eigenschaft gibt einen der folgenden Werte zurück:

- `ACTIVE`- Der iamRole ARN ist mit dem Cluster verknüpft und kann verwendet werden, um in Ihrem Namen auf andereAWS Dienste zuzugreifen.
- `PENDING`- Der IamRole-ARN wird dem Cluster zugeordnet.
- `INVALID`- Der iamRole-ARN ist dem Cluster zugeordnet, aber der Cluster kann die IAMRole nicht annehmen, um in Ihrem Namen auf andereAWS Dienste zuzugreifen.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++

- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBClusterSnapshot

Service: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem Cluster-Snapshot.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AvailabilityZones.AvailabilityZone.N

Stellt die Liste der Amazon EC2 Availability Zones bereit, in denen Instances im Cluster-Snapshot wiederhergestellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

ClusterCreateTime

Gibt den Zeitpunkt an, zu dem der Cluster erstellt wurde, in UTC (Universal Coordinated Time).

Typ: Zeitstempel

Erforderlich: Nein

DBClusterIdentifier

Gibt die Cluster-ID des Clusters an, aus dem dieser Cluster-Snapshot erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterSnapshotArn

Der Amazon-Ressourcenname (ARN) für den Cluster-Snapshot.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterSnapshotIdentifier

Gibt die Kennung für den Cluster-Snapshot an.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Legt den Namen der Datenbank-Engine fest.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Stellt die Version der Datenbank-Engine für diesen Cluster-Snapshot bereit.

Typ: Zeichenfolge

Erforderlich: Nein

KmsKeyId

Wenn `StorageEncrypted` ist`true`, die AWS KMS Schlüsselkennung für den verschlüsselten Cluster-Snapshot.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUsername

Gibt den Hauptbenutzernamen für den Cluster-Snapshot an.

Typ: Zeichenfolge

Erforderlich: Nein

PercentProgress

Gibt einen Prozentsatz der Daten an, die laut Schätzung bereits übertragen wurden.

Typ: Ganzzahl

Erforderlich: Nein

Port

Gibt den Port an, den der Cluster zum Zeitpunkt des Snapshots überwacht hat.

Typ: Ganzzahl

Erforderlich: Nein

SnapshotCreateTime

Gibt den Zeitpunkt an, zu dem der Snapshot erstellt wurde, in UTC.

Typ: Zeitstempel

Erforderlich: Nein

SnapshotType

Gibt den Typ des Cluster-Snapshots an.

Typ: Zeichenfolge

Erforderlich: Nein

SourceDBClusterSnapshotArn

Wenn der Cluster-Snapshot aus einem Quell-Cluster-Snapshot kopiert wurde, der ARN für den Quell-Cluster-Snapshot, andernfalls ein Nullwert.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Gibt den Status dieses Cluster-Snapshots an.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Gibt an, ob der Cluster-Snapshot verschlüsselt ist.

Typ: Boolesch

Erforderlich: Nein

StorageType

Speichertyp, der Ihrem Cluster-Snapshot zugeordnet ist

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB-Entwicklerhandbuch.

Gültige Werte für den Speichertyp – `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

VpcId

Gibt die Virtual Private Cloud (VPC)-ID an, die dem Cluster-Snapshot zugeordnet ist.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS-SDK für Ruby V3

DBClusterSnapshotAttribute

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält den Namen und die Werte eines manuellen Cluster-Snapshot-Attributs.

Manuelle Cluster-Snapshot-Attribute werden verwendet, um andereAWS-Konten zu autorisieren, einen manuellen Cluster-Snapshot wiederherzustellen.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AttributeName

Der Name des manuellen Cluster-Snapshot-Attributs.

Das benannte Attribut`restore` bezieht sich auf die Liste der PersonenAWS-Konten, die berechtigt sind, den manuellen Cluster-Snapshot zu kopieren oder wiederherzustellen.

Typ: Zeichenfolge

Required: No

AttributeValues.AttributeValue.N

Die Werte für das manuelle Cluster-Snapshot-Attribut.

Wenn das`AttributeName` Feld auf gesetzt ist`restore`, gibt dieses Element eine Liste der IDs der zurückAWS-Konten, die autorisiert sind, den manuellen Cluster-Snapshot zu kopieren oder wiederherzustellen. Wenn der Wert von in der Liste`all` steht, ist der manuelle Cluster-Snapshot öffentlich und kann von allenAWS-Konto kopiert oder wiederhergestellt werden.

Typ: Zeichenfolge-Array

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBClusterSnapshotAttributesResult

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu den Attributen, die einem Cluster-Snapshot zugeordnet sind.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterSnapshotAttributes.DBClusterSnapshotAttribute.N

Die Liste der Attribute und Werte für den Cluster-Snapshot.

Typ: Array von DBClusterSnapshotAttribute-Objekten

Required: No

DBClusterSnapshotIdentifier

Die ID des Cluster-Snapshots, für den die Attribute gelten.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBEngineVersion

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Engine-Version.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

DBEngineDescription

Die Beschreibung der Datenbank-Engine.

Typ: Zeichenfolge

Required: No

DBEngineVersionDescription

Die Beschreibung der Datenbank-Engine-Version.

Typ: Zeichenfolge

Required: No

DBParameterGroupFamily

Der Name der Parametergruppenfamilie für die Datenbank-Engine.

Typ: Zeichenfolge

Required: No

Engine

Der Name der Datenbank-Engine.

Typ: Zeichenfolge

Required: No

EngineVersion

Die Versionsnummer des Datenbank-Engines.

Typ: Zeichenfolge

Required: No

ExportableLogTypes.member.N

Die Protokolltypen, die die Datenbank-Engine für den Export nach Amazon CloudWatch Logs zur Verfügung stellt.

Typ: Zeichenfolge-Array

Required: No

SupportedCACertificateIdentifiers.member.N

Eine Liste der unterstützten CA-Zertifikatsbezeichner.

Weitere Informationen finden Sie unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten bei der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Typ: Zeichenfolge-Array

Required: No

SupportsCertificateRotationWithoutRestart

Gibt an, ob die Engine-Version das Rotieren des Serverzertifikats ohne Neustart der DB-Instance unterstützt.

Typ: Boolesch

Required: No

SupportsLogExportsToCloudwatchLogs

Ein Wert, der angibt, ob die Engine-Version das Exportieren der in `ExportableLogTypes` to CloudWatch Logs angegebenen Protokolltypen unterstützt.

Typ: Boolesch

Required: No

ValidUpgradeTarget.UpgradeTarget.N

Eine Liste der Engine-Versionen, auf die diese Datenbank-Engine-Version aktualisiert werden kann.

Typ: Array von UpgradeTarget-Objekten

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBInstance

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Instanz.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

AutoMinorVersionUpgrade

Trifft nicht zu. Dieser Parameter gilt nicht für Amazon DocumentDB. Amazon DocumentDB führt unabhängig vom eingestellten Wert keine kleineren Versions-Upgrades durch.

Typ: Boolesch

Required: No

AvailabilityZone

Gibt den Namen der Availability Zone an, in der sich die Instance befindet.

Typ: Zeichenfolge

Required: No

BackupRetentionPeriod

Gibt die Anzahl der Tage an, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Required: No

CACertificateIdentifier

Die ID des Zertifizierungsstellenzertifikats für diese DB-Instance.

Typ: Zeichenfolge

Required: No

CertificateDetails

Die Details des Serverzertifikats der DB-Instance.

Typ: CertificateDetails Objekt

Required: No

CopyTagsToSnapshot

Ein Wert, der angibt, ob Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden sollen. Standardmäßig werden Tags nicht kopiert.

Typ: Boolesch

Required: No

DBClusterIdentifier

Enthält den Namen des Clusters, dem die Instance angehört, wenn die Instance Mitglied eines Clusters ist.

Typ: Zeichenfolge

Required: No

DBInstanceArn

Der Amazon-Ressourcenname (ARN) für die Instance.

Typ: Zeichenfolge

Required: No

DBInstanceClass

Enthält den Namen der Rechen- und Speicherkapazitätsklasse der Instanz.

Typ: Zeichenfolge

Required: No

DBInstanceIdentifier

Enthält eine vom Benutzer bereitgestellte Datenbank-ID. Dieser Bezeichner ist der eindeutige Schlüssel, der eine Instanz identifiziert.

Typ: Zeichenfolge

Required: No

DBInstanceStatus

Gibt den aktuellen Status dieser Datenbank an.

Typ: Zeichenfolge

Required: No

DbiResourceId

Der AWS-Region -eindeutige, unveränderliche Bezeichner für die Instanz. Dieser Bezeichner befindet AWS CloudTrail sich in Protokolleinträgen, wenn auf den AWS KMS Schlüssel für die Instanz zugegriffen wird.

Typ: Zeichenfolge

Required: No

DBSubnetGroup

Gibt Informationen über die Subnetzgruppe an, die der Instance zugeordnet ist, einschließlich des Namens, der Beschreibung und der Subnetze in der Subnetzgruppe.

Typ: DBSubnetGroup Objekt

Required: No

EnabledCloudwatchLogsExports.member.N

Eine Liste der Protokolltypen, für deren Export diese Instanz in Logs konfiguriert ist. CloudWatch

Typ: Zeichenfolge-Array

Required: No

Endpoint

Gibt den Verbindungsendpunkt an.

Typ: Endpoint Objekt

Required: No

Engine

Gibt den Namen der Datenbank-Engine an, die für diese Instanz verwendet werden soll.

Typ: Zeichenfolge

Required: No

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Required: No

InstanceCreateTime

Gibt das Datum und die Uhrzeit der Erstellung der Instanz an.

Typ: Zeitstempel

Required: No

KmsKeyId

Falls `StorageEncrypted` ja`true`, die AWS KMS Schlüssel-ID für die verschlüsselte Instanz.

Typ: Zeichenfolge

Required: No

LatestRestorableTime

Gibt den letzten Zeitpunkt an, zu dem eine Datenbank mit point-in-time restore wiederhergestellt werden kann.

Typ: Zeitstempel

Required: No

PendingModifiedValues

Gibt an, dass Änderungen an der Instanz noch ausstehen. Dieses Element ist nur enthalten, wenn Änderungen ausstehen. Spezifische Änderungen werden von Unterelementen identifiziert.

Typ: PendingModifiedValues Objekt

Required: No

PreferredBackupWindow

Gibt den täglichen Zeitraum in koordinierter Weltzeit (UTC) an, in dem automatische Sicherungen erstellt werden, wenn automatische Sicherungen aktiviert sind, gemäß `BackupRetentionPeriod`.

Typ: Zeichenfolge

Required: No

PreferredMaintenanceWindow

Gibt den wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time) an.

Typ: Zeichenfolge

Required: No

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance heraufgestuft wird.

Typ: Ganzzahl

Required: No

PubliclyAccessible

Nicht unterstützt Amazon DocumentDB unterstützt derzeit keine öffentlichen Endgeräte. Der Wert von `PubliclyAccessible` ist immer. `false`

Typ: Boolesch

Required: No

StatusInfos.DBInstanceStatusInfo.N

Der Status einer Read Replica. Wenn es sich bei der Instanz nicht um ein Read Replica handelt, ist dieses Feld leer.

Typ: Array von DBInstanceStatusInfo-Objekten

Required: No

StorageEncrypted

Gibt an, ob die Instanz verschlüsselt ist oder nicht.

Typ: Boolesch

Required: No

VpcSecurityGroups.VpcSecurityGroupMembership.N

Stellt eine Liste der VPC-Sicherheitsgruppenelemente bereit, zu denen die Instance gehört.

Typ: Array von VpcSecurityGroupMembership-Objekten

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBInstanceStatusInfo

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt eine Liste mit Statusinformationen für eine Instanz bereit.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Message

Details des Fehlers, wenn ein Fehler bei der Instance auftritt. Wenn die Instance keinen Fehlerstatus aufweist, ist dieser Wert leer.

Typ: Zeichenfolge

Required: No

Normal

Ein boolescher Wert, der angibt,`true` ob die Instanz normal funktioniert oder`false` ob sich die Instanz in einem Fehlerzustand befindet.

Typ: Boolesch

Required: No

Status

Status der Instance. Für ein`StatusType` OF-Read-Replikat können die Werte`replicating`, error`stopped`, oder lauten`terminated`.

Typ: Zeichenfolge

Required: No

StatusType

Dieser Wert ist derzeit "`read replication`.“

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

DBSubnetGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Subnetzgruppe.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBSubnetGroupArn

Der Amazon-Ressourcenname (ARN) für die DB-Subnetzgruppe.

Typ: Zeichenfolge

Required: No

DBSubnetGroupDescription

Bietet die Beschreibung der Subnetzgruppe.

Typ: Zeichenfolge

Required: No

DBSubnetGroupName

Name der Subnetzgruppe.

Typ: Zeichenfolge

Required: No

SubnetGroupStatus

Liefert den Status der Subnetzgruppe.

Typ: Zeichenfolge

Required: No

Subnets.Subnet.N

Detaillierte Informationen zu einzelnen oder mehreren Subnetzen innerhalb einer Subnetzgruppe.

Typ: Array von Subnet-Objekten

Required: No

VpcId

Bietet die Virtual Private Cloud (VPC) ID der Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Endpoint

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Netzwerkinformationen für den Zugriff auf einen Cluster oder eine Instanz. Client-Programme müssen einen gültigen Endpunkt angeben, um auf diese Amazon DocumentDB DocumentDB-Ressourcen zugreifen zu können.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Address

Legt die DNS-Adresse der Instance fest.

Typ: Zeichenfolge

Required: No

HostedZoneId

Gibt die ID an, die Amazon Route 53 zuweist, wenn Sie eine gehostete Zone erstellen.

Typ: Zeichenfolge

Required: No

Port

Gibt die Portnummer an, die von der Datenbank-Engine überwacht wird.

Typ: Ganzzahl

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

EngineDefaults

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält das Ergebnis eines erfolgreichen Aufrufs derDescribeEngineDefaultClusterParameters Operation.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBParameterGroupFamily

Der Name der Cluster-Parametergruppenfamilie, für die die Cluster-Parametergruppe gibt die zurück gibt die Cluster-Parametergruppe zurück

Typ: Zeichenfolge

Required: No

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch MaxRecords angegebenen Wert.

Typ: Zeichenfolge

Required: No

Parameters.Parameter.N

Die Parameter einer bestimmten Cluster-Parametergruppenfamilie.

Typ: Array von Parameter-Objekten

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Event

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen über ein Ereignisereignis.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

Date

Legt das Datum und die Uhrzeit des Ereignisses fest.

Typ: Zeitstempel

Required: No

EventCategories.EventCategory.N

Legt die Kategorie für das Ereignis fest.

Typ: Zeichenfolge-Array

Required: No

Message

Stellt den Text dieses Ereignisses bereit.

Typ: Zeichenfolge

Required: No

SourceArn

Der Amazon-Ressourcenname (ARN) für das Ereignis.

Typ: Zeichenfolge

Required: No

SourceIdentifier

Stellt die Kennung für die Quelle des Ereignisses bereit.

Typ: Zeichenfolge

Required: No

SourceType

Gibt den Quelltyp für dieses Ereignis an.

Typ: Zeichenfolge

Zulässige Werte: `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

EventCategoriesMap

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein Ereignisquelltyp, begleitet von einem oder mehreren Namen der Ereigniskategorie.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

EventCategories.EventCategory.N

Die Ereigniskategorien für den angegebenen Quelltyp.

Typ: Zeichenfolge-Array

Required: No

SourceType

Der Quelltyp, zu dem die zurückgegebenen Kategorien gehören.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

EventSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

CustomerAwsId

DasAWS Kundenkonto, das mit dem Amazon DocumentDB DocumentDB-Abonnement für Eventbenachrichtigungen verknüpft ist.

Typ: Zeichenfolge

Required: No

CustSubscriptionId

Die Abonnement-ID für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Required: No

Enabled

Ein boolescher Wert, der angibt, ob das Abonnement aktiviert ist. Ein Wert von`true` gibt an, dass das Abonnement aktiviert ist.

Typ: Boolesch

Required: No

EventCategoriesList.EventCategory.N

Eine Liste der Ereigniskategorien für das Amazon DocumentDB DocumentDB-Abonnement für Ereignisbenachrichtigungen.

Typ: Zeichenfolge-Array

Required: No

EventSubscriptionArn

Der Amazon-Ressourcenname (ARN) für das Ereignisabonnement.

Typ: Zeichenfolge

Required: No

SnsTopicArn

Das Thema ARN des Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Required: No

SourceIdsList.SourceId.N

Eine Liste der Quell-IDs für das Amazon DocumentDB DocumentDB-Abonnement für Eventbenachrichtigungen.

Typ: Zeichenfolge-Array

Required: No

SourceType

Der Quelltyp für das Amazon DocumentDB DocumentDB-Abonnement für Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Required: No

Status

Der Status des Abonnements Amazon DocumentDB Ereignis-Benachrichtigung.

Einschränkungen:

Dies kann eine der folgenden Optionen sein:`creating`,`modifying`,`deleting`,`active`,`no-permission`,`topic-not-exist`

Derno-permission Status gibt an, dass keine Berechtigung mehr hat, an das Amazon-SNS-Thema zu posten. Dertopic-not-exist Status gibt an, dass das Thema gelöscht wurde, nachdem das Abonnement erstellt wurde.

Typ: Zeichenfolge

Required: No

SubscriptionCreationTime

Der Zeitpunkt, zu dem das Amazon DocumentDB Ereignis-Benachrichtigung erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Filter

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein benannter Satz von Filterwerten, der verwendet wird, um eine spezifischere Ergebnisliste zurückzugeben. Sie können einen Filter verwenden, um eine Reihe von Ressourcen nach bestimmten Kriterien, z. B. IDs, abzugleichen.

Platzhalter werden in Filtern nicht unterstützt.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Name

Der Name des Filters Bei Filternamen muss die Groß- und Kleinschreibung beachtet werden.

Typ: Zeichenfolge

Erforderlich: Ja

Values.Value.N

Ein oder mehrere Filterwerte. Bei Filterwerten muss die Groß- und Kleinschreibung beachtet werden.

Typ: Zeichenfolge-Array

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2

- AWS SDK für Ruby V3

GlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DatabaseName

Der Standard-Datenbankname innerhalb des neuen globalen Clusters.

Typ: Zeichenfolge

Required: No

DeletionProtection

Die Löschschutz-Einstellung für den neuen globalen -Cluster.

Typ: Boolesch

Required: No

Engine

Die Amazon DocumentDB DocumentDB-Datenbank-Engine, die vom globalen Cluster verwendet wird.

Typ: Zeichenfolge

Required: No

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Required: No

GlobalClusterArn

Der Amazon-Ressourcenname (ARN) für den globalen -Cluster.

Typ: Zeichenfolge

Required: No

GlobalClusterIdentifier

Enthält eine vom Benutzer bereitgestellte globale ClusterID. Diese ID ist der eindeutige Schlüssel zur Identifizierung eines globalen -Clusters.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge von 1. Höchstlänge = 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

GlobalClusterMembers.GlobalClusterMember.N

Die Liste der Cluster-IDs für sekundäre Cluster innerhalb des globalen Clusters. Derzeit auf einen Artikel limitiert.

Typ: Array von GlobalClusterMember-Objekten

Required: No

GlobalClusterResourceId

DieAWS-Region eindeutige, unveränderliche Kennung für den globalen Datenbank--Cluster. Diese ID wird inAWS CloudTrail Protokolleinträgen gefunden, wenn auf denAWS KMS Kundenmasterschlüssel (Customer Master Key, CMK) für den Cluster zugegriffen wird.

Typ: Zeichenfolge

Required: No

Status

Gibt den aktuellen Status dieses globalen -Clusters an.

Typ: Zeichenfolge

Required: No

StorageEncrypted

Die Speicherverschlüsselung für den globalen -Cluster.

Typ: Boolesch

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

GlobalClusterMember

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Eine Datenstruktur mit Informationen über alle primären und sekundären Cluster, die einem globalen Amazon DocumentDB-Cluster zugeordnet sind.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterArn

Der Amazon-Ressourcenname (ARN) für jeden Amazon-DokumentDB-Cluster.

Typ: Zeichenfolge

Required: No

IsWriter

Gibt an, ob der Amazon DocumentDB-Cluster der primäre Cluster ist (d. h. er verfügt über Lese- und Schreibfunktionen) für den globalen Amazon DocumentDB-Cluster, dem er zugeordnet ist.

Typ: Boolesch

Required: No

Readers.member.N

Der Amazon-Ressourcenname (ARN) für jeden schreibgeschützten sekundären Cluster, der dem DB-Cluster zugeordnet ist.

Typ: Zeichenfolge-Array

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

OrderableDBInstanceOption

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Die Optionen, die für eine Instanz verfügbar sind.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AvailabilityZones.AvailabilityZone.N

Eine Liste der Availability Zones für eine Instance.

Typ: Array von AvailabilityZone-Objekten

Required: No

DBInstanceClass

Die Instanzklasse für eine Instanz.

Typ: Zeichenfolge

Required: No

Engine

Der Engine-Typ einer Instanz.

Typ: Zeichenfolge

Required: No

EngineVersion

Die Engine-Version einer Instanz.

Typ: Zeichenfolge

Required: No

LicenseModel

Das Lizenzmodell für eine Instance.

Typ: Zeichenfolge

Required: No

Vpc

Gibt an, ob sich eine Instance in einer Virtual Private Cloud (VPC) befindet.

Typ: Boolesch

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Parameter

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem einzelnen Parameter.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

AllowedValues

Gibt den gültigen Wertebereich für den Parameter an.

Typ: Zeichenfolge

Required: No

ApplyMethod

Gibt an, wann Parameteraktualisierungen angewendet werden können.

Typ: Zeichenfolge

Zulässige Werte: `immediate | pending-reboot`

Required: No

ApplyType

Gibt den Engine-spezifischen Parametertyp an.

Typ: Zeichenfolge

Required: No

DataType

Gibt den gültigen Datentyp für den Parameter an.

Typ: Zeichenfolge

Required: No

Description

Stellt eine Beschreibung des Parameters bereit.

Typ: Zeichenfolge

Required: No

IsModifiable

Gibt an, ob der Parameter geändert werden kann oder nicht (`true` oder `false`). Einige Parameter wirken sich auf die Sicherheit oder die betrieblichen Abläufe aus und können nicht geändert werden.

Typ: Boolesch

Required: No

MinimumEngineVersion

Die älteste Engine-Version, auf die der Parameter angewendet werden kann.

Typ: Zeichenfolge

Required: No

ParameterName

Gibt den Namen des Parameters an.

Typ: Zeichenfolge

Required: No

ParameterValue

Gibt den Wert des Parameters an.

Typ: Zeichenfolge

Required: No

Source

Gibt die Quelle des Parameterwerts an.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK für Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

PendingCloudwatchLogsExports

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Eine Liste der Protokolltypen, deren Konfiguration noch aussteht. Diese Flotten werden gerade erfüllt.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

LogTypesToDisable.member.N

Protokolltypen, die gerade aktiviert werden. Nach ihrer Aktivierung werden diese Protokolltypen nach Amazon CloudWatch Logs exportiert.

Typ: Zeichenfolge-Array

Required: No

LogTypesToEnable.member.N

Protokolltypen, die gerade deaktiviert werden. Nach ihrer Deaktivierung werden diese Protokolltypen nicht in CloudWatch Logs exportiert.

Typ: Zeichenfolge-Array

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

PendingMaintenanceAction

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt Informationen über eine ausstehende Wartungsaktion für eine Ressource bereit.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Action

Der Typ der ausstehenden Wartungsaktion, die für die Ressource verfügbar ist.

Typ: Zeichenfolge

Required: No

AutoAppliedAfterDate

Das Datum des Wartungsfensters, in dem die Aktion angewendet wird. Die Wartungsaktion wird während ihres ersten Wartungsfensters nach diesem Datum auf die Ressource angewendet. Wenn dieses Datum angegeben ist, werden alle `next-maintenance`-Opt-In-Anfragen ignoriert.

Typ: Zeitstempel

Required: No

CurrentApplyDate

Das Datum des Inkrafttretens, wenn die ausstehende Wartungsaktion auf die Ressource angewendet wird.

Typ: Zeitstempel

Required: No

Description

Eine Beschreibung, die weitere Details zu der Wartungsaktion bereitstellt.

Typ: Zeichenfolge

Required: No

ForcedApplyDate

Das Datum, an dem die Wartungsaktion automatisch angewendet wird. Die Wartungsaktion wird ungeachtet des Wartungsfensters für die Ressource an diesem Datum auf die Ressource angewendet. Wenn dieses Datum angegeben ist, werden alle `immediate`-Opt-In-Anfragen ignoriert.

Typ: Zeitstempel

Required: No

OptInStatus

Gibt den Typ der Opt-in-Anforderung an, die für die Ressource empfangen wurde.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

PendingModifiedValues

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Eine oder mehrere geänderte Einstellungen für eine Instanz. Diese geänderten Einstellungen wurden angefordert, aber noch nicht angewendet.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AllocatedStorage

Enthält die neue`AllocatedStorage` Größe für die Instanz, die angewendet wird oder gerade angewendet wird.

Typ: Ganzzahl

Required: No

BackupRetentionPeriod

Gibt die ausstehende Anzahl von Tagen an, die automatische Sicherungen aufbewahrt werden.

Typ: Ganzzahl

Required: No

CACertificateIdentifier

Gibt die ID des Zertifizierungsstelle (CA) Zertifizierungsstelle (CA) für die DB-Instance an.

Typ: Zeichenfolge

Required: No

DBInstanceClass

Enthält das Neue`DBInstanceClass` für die Instanz, das angewendet wird oder gerade angewendet wird.

Typ: Zeichenfolge

Required: No

DBInstanceIdentifier

Enthält das Neue`DBInstanceIdentifier` für die Instanz, das angewendet wird oder gerade angewendet wird.

Typ: Zeichenfolge

Required: No

DBSubnetGroupName

Die neue Subnetzgruppe für die Instance.

Typ: Zeichenfolge

Required: No

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Required: No

Iops

Gibt den neuen Provisioned IOPS-Wert für die Instanz an, die angewendet wird oder gerade angewendet wird.

Typ: Ganzzahl

Required: No

LicenseModel

Das Lizenzmodell für die Instanz.

Zulässige Werte: `license-included`, `bring-your-own-license`, `general-public-license`

Typ: Zeichenfolge

Required: No

MasterUserPassword

Enthält die ausstehende oder derzeit laufende Änderung der Master-Anmeldeinformationen für die Instance.

Typ: Zeichenfolge

Required: No

MultiAZ

Gibt an, dass die Single-AZ-Instance zu einer Multi-AZ-Bereitstellung wechseln.

Typ: Boolesch

Required: No

PendingCloudwatchLogsExports

Eine Liste der Protokolltypen, deren Konfiguration noch aussteht. Diese Protokolltypen werden gerade aktiviert oder deaktiviert.

Typ: PendingCloudwatchLogsExports Objekt

Required: No

Port

Gibt den ausstehenden Port für die Instanz an.

Typ: Ganzzahl

Required: No

StorageType

Gibt den Speichertyp an, der der Instance zugeordnet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

ResourcePendingMaintenanceActions

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt die Ausgabe von darApplyPendingMaintenanceAction.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

PendingMaintenanceActionDetails.PendingMaintenanceAction.N

Eine Liste mit Details zu den ausstehenden Wartungsaktionen für die Ressource.

Typ: Array von PendingMaintenanceAction-Objekten

Required: No

ResourceIdentifier

Der Amazon-Ressourcenname (ARN) der Ressource.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Subnet

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

SubnetAvailabilityZone

Detaillierte

Typ: AvailabilityZone Objekt

Required: No

SubnetIdentifier

Gibt die Kennung des Subnetzes an.

Typ: Zeichenfolge

Required: No

SubnetStatus

Gibt den Status des Subnetzes an.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go

- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Tag

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Metadaten, die aus einem Schlüssel-Paar Schlüssel-Paar Amazon DocumentDB Schlüssel-Paar Schlüssel-Paar Schlüssel-Paar Schlüssel-Paar Schlüssel-Paar Schlüssel

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Key

Schlüsselname Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein „" "“ oder „`aws:"rds:`“ als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie „_“, „Leerzeichen“, „_“, „Leerzeichen“, „_“, „Leerzeichen“, „ . „/“, „=“, „+“, „-“ (Java-Regex: „^ ([\\ p {L}\\ p {L}\\ p {L}\\ p {L}\\ p {N} _.: /=+\\ -] *)“).

Typ: Zeichenfolge

Required: No

Value

Schlüsselname Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Tags Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein „" "“ oder „`aws:"rds:`“ als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen sowie „_“, „Leerzeichen“, „_“, „Leerzeichen“, „_“, „Leerzeichen“, „ . „/“, „=“, „+“, „-“ (Java-Regex: „^ ([\\ p {L}\\ p {L}\\ p {L}\\ p {L}\\ p {N} _.: /=+\\ -] *)“).

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

UpgradeTarget

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Die Version der Datenbank-Engine, auf die eine Instanz aktualisiert werden kann.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AutoUpgrade

Ein Wert, der angibt, ob die Zielversion auf alle Quell-DB-Instances angewendet wird, die auf`AutoMinorVersionUpgrade` gesetzt haben`true`.

Typ: Boolesch

Required: No

Description

Die Version der Datenbank-Engine, auf die eine Instanz aktualisiert werden kann.

Typ: Zeichenfolge

Required: No

Engine

Der Name der Upgrade-Zieldatenbank-Engine.

Typ: Zeichenfolge

Required: No

EngineVersion

Die Versionsnummer der Upgrade-Zieldatenbank-Engine.

Typ: Zeichenfolge

Required: No

IsMajorVersionUpgrade

Ein Wert, der angibt, ob eine Datenbank-Engine auf eine Hauptversion aktualisiert wird.

Typ: Boolesch

Required: No

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

VpcSecurityGroupMembership

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Wird als Antwortelement für Anfragen zur VPC-Sicherheitsgruppe (Virtual Private Cloud) für Cloud (VPC) -Sicherheitsgruppe verwendet.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Status

Der Status der VPC-Sicherheitsgruppe.

Typ: Zeichenfolge

Required: No

VpcSecurityGroupId

Der Name der VPC-Sicherheitsgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Amazon DocumentDB Elastic Clusters

Die folgenden Datentypen werden von Amazon DocumentDB Elastic Clustern unterstützt:

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- Shard
- ValidationExceptionField

Cluster

Service: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu einem bestimmten elastischen Cluster zurück.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

adminUserName

Der Name des Elastic-Cluster-Administrators.

Typ: Zeichenfolge

Erforderlich: Ja

authType

Der Authentifizierungstyp für den elastischen Cluster.

Typ: Zeichenfolge

Zulässige Werte: `PLAIN_TEXT | SECRET_ARN`

Erforderlich: Ja

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

clusterEndpoint

Die URL, die für die Verbindung mit dem elastischen Cluster verwendet wird.

Typ: Zeichenfolge

Erforderlich: Ja

clusterName

Der Name des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

createTime

Der Zeitpunkt, zu dem der elastische Cluster in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

kmsKeyId

Die KMS-Schlüsselkennung, die zum Verschlüsseln des elastischen Clusters verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

preferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Typ: Zeichenfolge

Erforderlich: Ja

shardCapacity

Die Anzahl der vCPUs, die jedem elastischen Cluster-Shard zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.

Typ: Ganzzahl

Erforderlich: Ja

shardCount

Die Anzahl der dem elastischen Cluster zugewiesenen Shards. Das Maximum ist 32.

Typ: Ganzzahl

Erforderlich: Ja

status

Der Status des elastischen Clusters.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Erforderlich: Ja

subnetIds

Die Amazon EC2-Subnetz-IDs für den elastischen Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

vpcSecurityGroupIds

Eine Liste der EC2-VPC-Sicherheitsgruppen, die mit diesem elastischen Cluster verknüpft sind.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

backupRetentionPeriod

Die Anzahl der Tage, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

preferredBackupWindow

Der tägliche Zeitraum, in dem automatisierte Backups erstellt werden, wenn automatisierte Backups aktiviert sind, wie von festgelegt`backupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikat-Instances, die für alle Shards im Cluster gelten. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt, und alle zusätzlichen Instances sind Replikate, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

shards

Die Gesamtzahl der Shards im Cluster.

Typ: Array von Shard-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für Ruby V3

ClusterInList

Service: Amazon DocumentDB Elastic Clusters

Eine Liste der elastischen Amazon DocumentDB-Cluster.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

clusterName

Der Name des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der Status des elastischen Clusters.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie im Folgenden:

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für Ruby V3

ClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu einem bestimmten Elastic-Cluster-Snapshot zurück.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

adminUserName

Der Name des Elastic-Cluster-Administrators.

Typ: Zeichenfolge

Erforderlich: Ja

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

clusterCreationTime

Der Zeitpunkt, zu dem der elastische Cluster in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

kmsKeyId

Die Kennung für den KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon-Konto erstellen, das diesen KMS-Verschlüsselungsschlüssel besitzt, können Sie den KMS-Schlüsselalias anstelle des ARN als KMS-Verschlüsselungsschlüssel verwenden. Wenn hier kein Verschlüsselungsschlüssel angegeben ist, verwendet Amazon DocumentDB den Standardverschlüsselungsschlüssel,

den KMS für Ihr Konto erstellt. Ihr Konto verfügt für jede Amazon-Region über einen anderen Standardverschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotArn

Die ARN-ID des elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotCreationTime

Der Zeitpunkt, zu dem der elastische Cluster-Snapshot in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotName

Der Name des elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der Status des elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Erforderlich: Ja

subnetIds

Die Amazon EC2-Subnetz-IDs für den elastischen Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

vpcSecurityGroupIds

Eine Liste der EC2-VPC-Sicherheitsgruppen, die dem elastischen Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

snapshotType

Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:

- `automated` – Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr AWS Konto erstellt hat.
- `manual` – Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr AWS Konto erstellt haben.

Typ: Zeichenfolge

Zulässige Werte: `MANUAL | AUTOMATED`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für Ruby V3

ClusterSnapshotInList

Service: Amazon DocumentDB Elastic Clusters

Eine Liste von Elastic-Cluster-Snapshots.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotArn

Die ARN-ID des elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotCreationTime

Der Zeitpunkt, zu dem der elastische Cluster-Snapshot in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotName

Der Name des elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der Status des elastischen Cluster-Snapshots.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für Ruby V3

Shard

Service: Amazon DocumentDB Elastic Clusters

Der Name des Shards.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

createTime

Der Zeitpunkt, zu dem der Shard in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

shardId

Die ID des Shards.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der aktuelle Status des Shards.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS SDKs finden Sie unter:

- AWS SDK für C++
- AWS SDK für Go
- AWS SDK für Java V2
- AWS SDK für Ruby V3

ValidationExceptionField

Bedienung:Amazon DocumentDB Elastic Clusters

Ein bestimmtes Feld, in dem eine bestimmte Überprüfungsausnahme aufgetreten ist.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

message

Eine Fehlermeldung, die die Validierungsausnahme in diesem Feld beschreibt.

Typ: Zeichenfolge

Erforderlich: Ja

name

Der Name des Felds, in dem die Überprüfungsausnahme aufgetreten ist.

Typ: Zeichenfolge

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einem der sprachspezifischen AWS-SDKs finden Sie unter:

- AWS-SDK für C++
- AWS-SDK for Go
- AWS-SDK für Java V2
- AWS SDK für Ruby V3

Häufige Fehler

In diesem Abschnitt sind Fehler aufgeführt, die häufig bei den API-Aktionen aller AWS-Services auftreten. Informationen zu Fehlern, die spezifisch für eine API-Aktion für diesen Service sind, finden Sie unter dem Thema für diese API-Aktion.

AccessDeniedException

Sie haben keinen ausreichenden Zugriff zum Durchführen dieser Aktion.

HTTP Status Code: 400

IncompleteSignature

Die Anforderungssignatur entspricht nicht den AWS-Standards.

HTTP Status Code: 400

InternalFailure

Die Anforderungsverarbeitung ist fehlgeschlagen, da ein unbekannter Fehler, eine Ausnahme oder ein Fehler aufgetreten ist.

HTTP Status Code: 500

InvalidAction

Die angeforderte Aktion oder Operation ist ungültig. Überprüfen Sie, ob die Aktion ordnungsgemäß eingegeben wurde.

HTTP Status Code: 400

InvalidClientTokenId

Das angegebene X.509-Zertifikat oder die AWS-Zugriffsschlüssel-ID ist nicht in unseren Datensätzen vorhanden.

HTTP Status Code: 403

NotAuthorized

Sie haben keine Berechtigung zum Ausführen dieser Aktion.

HTTP Status Code: 400

OptInRequired

Die AWS-Zugriffsschlüssel-ID benötigt ein Abonnement für den Service.

HTTP Status Code: 403

RequestExpired

Die Anforderung hat den Service mehr als 15 Minuten nach dem Datumsstempel oder mehr als 15 Minuten nach dem Ablaufdatum der Anforderung erreicht (z. B. für vorsignierte URLs) oder der Datumsstempel auf der Anforderung liegt mehr als 15 Minuten in der Zukunft.

HTTP Status Code: 400

ServiceUnavailable

Die Anforderung ist aufgrund eines temporären Fehlers des Servers fehlgeschlagen.

HTTP Status Code: 503

ThrottlingException

Die Anforderung wurde aufgrund der Drosselung von Anforderungen abgelehnt.

HTTP Status Code: 400

ValidationError

Die Eingabe erfüllt nicht die von einem AWS-Service definierten Einschränkungen.

HTTP Status Code: 400

Geläufige Parameter

Die folgende Liste enthält die Parameter, die alle Aktionen zum Signieren von Signature-Version-4-Anforderungen mit einer Abfragezeichenfolge verwenden. Alle aktionsspezifischen Parameter werden im Thema für diese Aktion aufgelistet. Weitere Informationen zu Signature Version 4 finden Sie unter Signieren vonAWS API-Anfragen im IAM-Benutzerhandbuch.

Action

Die auszuführende Aktion.

Typ: Zeichenfolge

Erforderlich: Ja

Version

Die API-Version, für die die Anforderung geschrieben wurde, ausgedrückt im Format JJJJ-MM-TT.

Typ: Zeichenfolge

Erforderlich: Ja

X-Amz-Algorithm

Der Hashalgorithmus, den Sie zum Erstellen der Anforderungssignatur verwendet haben.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Zulässige Werte: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

Der Wert des Anmeldeinformationsumfangs. Dabei handelt es sich um eine Zeichenfolge, die Ihren Zugriffsschlüssel, das Datum, die gewünschte Region und eine Zeichenfolge zur Beendigung („aws4_request“) beinhaltet. Der Wert wird im folgenden Format ausgedrückt: Zugriffsschlüssel/JJJJMMTT/Region/Service/aws4_request.

Weitere Informationen finden Sie unter Erstellen einer signiertenAWS API-Anfrage im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Date

Das Datum, das zum Erstellen der Signatur verwendet wird. Das Format muss das ISO 8601-Basisformat (JJJJMMTT'T'SSMMSS'Z') sein. Die folgende Datumszeit ist beispielsweise ein gültiger X-Amz-Date-Wert: `20120325T120000Z`.

Bedingung: X-Amz-Date ist bei allen Anforderungen optional. Damit kann das Datum überschrieben werden, das zum Signieren von Anforderungen verwendet wird. Wenn der Date-Header im ISO 8601-Basisformat angegeben ist, ist X-Amz-Date nicht erforderlich. Wenn X-Amz-Date verwendet wird, überschreibt es immer den Wert des Date-Headers. Weitere Informationen finden Sie unter Elemente einerAWS API-Anforderungssignatur im IAM-Benutzerhandbuch.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Security-Token

Das temporäre Sicherheitstoken, das durch einen Anruf vonAWS Security Token Service (AWS STS). Eine Liste der Services, die temporäre Sicherheitsanmeldeinformationen von unterstützenAWS STS AWS-Services, finden Sie unter, die mit IAM arbeiten im IAM-Benutzerhandbuch.

Bedingung: Wenn Sie temporäre Sicherheitsanmeldeinformationen von nutzenAWS STS, müssen Sie das Sicherheitstoken einschließen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Signature

Gibt die hex-codierte Signatur an, die aus der zu signierenden Zeichenfolge und dem abgeleiteten Signaturschlüssel berechnet wurde.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-SignedHeaders

Gibt alle HTTP-Header an, die als Teil der kanonischen Anforderung enthalten waren. Weitere Informationen zur Angabe signierter Header finden Sie unter Erstellen einer signiertenAWS API-Anfrage im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

Versionshinweise

In diesen Versionshinweisen werden die Funktionen, Verbesserungen und Fehlerbehebungen von Amazon DocumentDB nach Veröffentlichungsdatum beschrieben. Die Versionshinweise enthalten Updates für alle Amazon DocumentDB-Engine-Versionen, sobald sie auftreten.

Sie können die aktuelle Patch-Version der Amazon DocumentDB-Engine ermitteln, indem Sie den folgenden Befehl ausführen:

```
db.runCommand({getEngineVersion: 1})
```

Wenn Ihr Cluster nicht auf der neuesten Version der Engine läuft, ist es wahrscheinlich, dass Sie über ausstehende Wartungsarbeiten verfügen, die Ihre Engine aktualisieren. Weitere Informationen finden Sie unter Verwalten von Amazon DocumentDB im -Entwicklerhandbuch.

Themen

22. Februar 2024

Neue Features

Elastische Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster unterstützen jetzt die folgenden Funktionen:

- Readable Secondary Shard Instance Replicas – weitere Informationen finden Sie in Schritt 5b von Schritt 1: Erstellen eines elastischen Clusters.
- Cluster starten/stoppen – Weitere Informationen finden Sie unter Stoppen und Starten eines elastischen Amazon DocumentDB-Clusters.
- Konfigurierbare Shard-Instances – weitere Informationen finden Sie in Schritt 5b von Schritt 1: Erstellen eines elastischen Clusters.
- Automatische Backups für Snapshots – weitere Informationen finden Sie unter Verwalten eines automatischen Backups für Elastic-Cluster-Snapshots.
- Snapshot kopieren – weitere Informationen finden Sie unter Kopieren eines Elastic-Cluster-Snapshots.

30. Januar 2024

Neue Features

Elastische Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster sind jetzt in den folgenden Regionen verfügbar:

- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Seoul)
- Südamerika (São Paulo)
- Europa (London)

Weitere Informationen finden Sie unter Verfügbarkeit von Elastic Cluster-Regionen und -Versionen.

Globale Amazon DocumentDB-Cluster

Globale Cluster sind jetzt in beiden AWS GovCloud (US) Regionen verfügbar: AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West).

10. Januar 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Versionen 3.0.4574, 3.0.4780, 3.0.4960)

- Unterstützung für SW-Vektorindizes hinzugefügt. Weitere Informationen finden Sie unter Vektorsuche nach Amazon DocumentDB.
- Unterstützung für partielle Indizes hinzugefügt. Weitere Informationen finden Sie unter Teilweiser Index.
- Unterstützung für die GC-Laufzeit für eine Sammlung innerhalb des `currentOp` Befehls hinzugefügt.
- Unterstützung des Textindex für die native Textsuche in Amazon DocumentDB hinzugefügt. Weitere Informationen finden Sie unter Durchführen einer Textsuche mit Amazon DocumentDB.
- Unterstützung für Schema`$jsonSchema`-Schlüsselwörter `type`, `allOf`, `oneOf`, `anyOf`, `not`, `maxItems`, `minItems`, `maxProperties`, `minProperties`, , `pattern`, `patternProperties`, `multipleOfdependencies`, und hinzugefügt`uniqueItems`.

  Weitere Informationen finden Sie unter Verwenden der JSON-Schemavalidierung.
- Unterstützung für arithmetische Operatoren `$ceil`, `$floor`, `$ln`, `$log`, `$log10``$sqrt`, und hinzugefügt`$exp`.

  Weitere Informationen finden Sie unter Arithmetische Operatoren.
- Unterstützung für parallele `IVFFLAT` Vektorindex-Builds hinzugefügt. Die Dokumentation wurde aktualisiert, indem die `IVFFLAT` Parallelvektorindex-Builds-Beschränkung aus dem Entwicklerhandbuch entfernt wurde.

Amazon DocumentDB 4.0 (Engine-Patch-Versionen 2.0.10124, 2.0.10179, 2.0.10221)

- Unterstützung für die GC-Laufzeit für eine Sammlung innerhalb des `currentOp` Befehls hinzugefügt.
- Unterstützung für Schema`$jsonSchema`-Schlüsselwörter `type`, `allOf`, `oneOf`, `anyOf`, `not`, `maxItems`, `minItems`, `maxProperties`, `minProperties`, , `pattern`, `patternProperties`, `multipleOfdependencies`, und hinzugefügt`uniqueItems`.

  Weitere Informationen finden Sie unter Verwenden der JSON-Schemavalidierung.

- Unterstützung für arithmetische Operatoren `$ceil`, `$floor`, `$ln`, `$log`, `$log10$sqrt`, und hinzugefügt`$exp`.

  Weitere Informationen finden Sie unter Arithmetische Operatoren.

Fehlerbehebungen und andere Änderungen

- Es wurde eine Funktion ohne Berücksichtigung der Groß- und Kleinschreibung für den Aufruf von hinzugefügt`db.runCommand("dbstats")`. Kunden von Amazon DocumentDB 5.0 und 4.0, die Engine-Patch-Versionen vor 3.0.4960 oder 2.0.10221 verwenden, sollten diese neuesten Engine-Patches anwenden.
- Es wurde ein Fehler beim Aufrufen `db.coll.stats()` von auf mongo-Shell-Version 1.7 und höher behoben. Die Dokumentation wurde aktualisiert, indem der Tipp `db.coll.stats()` zur Fehlerbehebung bei der Mongo-Shell aus dem Entwicklerhandbuch entfernt wurde.

20. Dezember 2023

Weitere Änderungen

Unterstützung für direkte Hauptversions-Upgrades in Amazon DocumentDB 3.6 und 4.0 aktiviert. Weitere Informationen finden Sie unter Direktes Upgrade der Hauptversion von Amazon DocumentDB .

13. Dezember 2023

Neue Features

Unterstützung für EC2-Konnektivität mit einem Klick hinzugefügt. Weitere Informationen finden Sie unter Herstellen einer Verbindung über Amazon EC2.

29. November 2023

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3727)

Neue Features

Unterstützung für die Vektorsuche hinzugefügt. Weitere Informationen finden Sie in diesem Blogbeitrag und unter Vektorsuche nach Amazon DocumentDB im Amazon DocumentDB-Entwicklerhandbuch.

21. November 2023

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3727)

Neue Features

Unterstützung für E/A-optimierten Speicher hinzugefügt. Weitere Informationen finden Sie unter Speicherkonfigurationen für Amazon DocumentDB-Cluster im Amazon DocumentDB-Entwicklerhandbuch.

Integration für Machine Learning ohne Code mit SageMaker Canvas hinzugefügt. Weitere Informationen finden Sie unter Machine Learning ohne Code mit Amazon SageMaker Canvas im Amazon DocumentDB-Entwicklerhandbuch.

17. November 2023

Neue Features

Amazon DocumentDB ist jetzt in der Region AWS GovCloud (USA-Ost) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208570)

Benutzerdefinierte lokale Variablennamen unterstützen jetzt „_“ (Unterstrich) für Projektionsoperatoren wie `$let` und `$filter`.

6. November 2023

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3727) und 4.0 (Engine-Patch-Version 2.0.9876)

Neue Features

- Unterstützung für Schema`$jsonSchema`-Schlüsselwörter `maxLength`, `minLength`, `maximum`, `minimum`, `exclusiveMaximum`, `items`, `exclusiveMinimum`und hinzugefügt`additionalItems`.

  Bitte beachten Sie, dass die JSON-Schemavalidierung nur in Instance-basierten Clustern unterstützt wird.
- Unterstützung für `$convert` den Aggregations-Pipeline-Operator und seine abgeleiteten Kurznotationen `$toBool`, `$toInt`, `$toLong`, `$toDouble`, `$toDecimal`, `$toString``$toObjectId`, und hinzugefügt`$toDate`.
- Unterstützung für Satzausdrucksoperatoren `$setDifference`, `$anyElementTrue`und hinzugefügt`$allElementTrue`.

Fehlerbehebungen und andere Änderungen

Es wurde ein Problem behoben, bei dem ein Change-Stream-Update von `-NaN` auf nicht angezeigt `NaN` wurde.

20. Oktober 2023

Weitere Änderungen

Amazon DocumentDB hat ein Problem identifiziert und verbietet vorübergehend Hauptversions-Upgrades (MVU) in allen Regionen. Wir haben die Ursache für das Problem identifiziert und eine Lösung entwickelt, die derzeit getestet wird. Wir gehen davon aus, dass dieser Patch vor Ende des Q4 2023 in allen Regionen bereitgestellt wird. MVU bleibt deaktiviert, bis der Fix in allen Regionen bereitgestellt wird. Weitere Aktualisierungen zur Verfügbarkeit von MVU-Features finden Sie auf dieser Versionshinweisseite.

In der Zwischenzeit können Sie verwenden, AWS DMS um Hauptversions-Upgrades durchzuführen, indem Sie Ihre Amazon DocumentDB-Datenbank von einem Cluster mit niedrigerer Version auf eine höhere Version migrieren. Führen Sie die Schritte unter ausAktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service, um ein Upgrade mit durchzuführen AWS DMS. In diesem Blogbeitrag finden Sie weitere Informationen zu bewährten Methoden, die Sie beim Upgrade mit befolgen sollten AWS DMS.

25. September 2023

Neue Features

Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Hongkong) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

20. September 2023

Neue Features

Unterstützung für direkte Hauptversions-Upgrades in Amazon DocumentDB 3.6 und 4.0 hinzugefügt. Weitere Informationen finden Sie unter Direktes Upgrade der Hauptversion von Amazon DocumentDB .

15. September 2023

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3140) und 4.0 (Engine-Patch-Version 2.0.9686)

- Unterstützung für die $jsonSchema-Schemavalidierung nur in Instance-basierten Clustern hinzugefügt.

  Weitere Informationen finden Sie unter Verwenden der JSON-Schemavalidierung.

11. September 2023

Neue Features

Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Hyderabad) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

3. August 2023

Neue Features

Amazon DocumentDB-Elastic-Cluster

- Amazon DocumentDB-Elastic-Cluster unterstützen jetzt die folgenden Operationen:
  - `top`
  - `collStats`
  - `hint`
  - `dataSize`

  Eine vollständige Liste der unterstützten Befehle und Operationen Unterstützte MongoDB-APIs, -Operationen und -Datentypen finden Sie unter .
- Time to Live (TTL)-Indizes werden jetzt unterstützt.
- Index `hints` wird jetzt mit Indexausdrücken unterstützt.

13. Juli 2023

Neue Funktionen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.1948)

- Unterstützung für die Dokumentkomprimierung hinzugefügt.
- Unterstützung für parallele Index-Builds hinzugefügt.
- Unterstützung für den Index-Build-Status hinzugefügt.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.9259)

- Unterstützung für parallele Index-Builds hinzugefügt.

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.1948)

- Es wurde ein Authentifizierungsproblem mit `createCollection` für elastische Amazon DocumentDB-Cluster behoben, wenn Benutzer keinen Zugriff auf Systemsammlungen haben.
- Es wurde ein Problem behoben, bei dem Instances der sekundären Region nicht dieselben Instance-Namen der primären Region verwenden konnten.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.9259)

- Das Hinzufügen interner Überwachungsabfragen zu Prüfungsprotokollen wurde gestoppt.

7. Juni 2023

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 5.0

- r5- und t3.medium-Instances werden jetzt in Amazon DocumentDB 5.0 unterstützt.
- `engineVersion` Die Standardoption ist `5.0.0` in AWS SDK AWS CLI, und AWS CloudFormation.

10. Mai 2023

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.1361)

- Unterstützung für `ignoreunknownindexoptions` im createIndex Befehl hinzugefügt.
- Das Hinzufügen interner Überwachungsabfragen zu Prüfungsprotokollen wurde gestoppt.
- Benutzerdefinierte lokale Variablennamen unterstützen jetzt „_“ (Unterstrich) für Projektionsoperatoren wie `$let` und `$filter`.

4. April 2023

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.8934)

- Es wurde ein Problem mit der DML-Prüfung behoben, wenn sie während eines laufenden Workloads aktiviert ist.
- Es wurde ein Problem mit der DML-Prüfung behoben, wenn Aggregatbefehle mit Hinweis einen Zeichenfolgenwert übergeben.
- Es wurde ein Problem behoben, bei dem `listCollections` der Befehl nicht funktionierte, wenn Benutzer mit readwriteanydatabase-Rolle sowohl authorizedCollections als auch nameOnly-Optionen auf „true“ gesetzt waren.

- Es wurde ein Problem behoben, bei dem numerische Zeichenfolgen in einem Feldnamen korrekt analysiert wurden.
- Stornieren Sie lange laufende Cursor, wenn sie sich auf die Garbage Collection auswirken.
- Benutzerdefinierte lokale Variablennamen unterstützen jetzt „_“ (Unterstrich) für Projektionsoperatoren wie `$let` und `$filter`.

22. März 2023

Neue Features

Elastische Amazon DocumentDB-Cluster sind jetzt in den Regionen Asien-Pazifik (Singapur), Asien-Pazifik (Sydney) und Asien-Pazifik (Tokio) verfügbar. Weitere Informationen finden Sie unter Verfügbarkeit von Elastic Cluster-Regionen und -Versionen.

1. März 2023

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.775)

- Einführung von Amazon DocumentDB 5.0
  - MongoDB 5.0-Kompatibilität (Unterstützung für MongoDB 5.0-API-Treiber)
  - Unterstützung für die clientseitige Verschlüsselung auf Feldebene (FLE). Sie können jetzt Felder clientseitig verschlüsseln, bevor Sie die Daten in den Amazon DocumentDB-Cluster schreiben. Weitere Informationen finden Sie unter Clientseitige Verschlüsselung auf Feldebene.
  - Neue Aggregationsoperatoren: `$dateAdd`, `$dateSubtract`
- Das Speicherlimit für alle Instance-basierten Amazon DocumentDB-Cluster und Shard-basierten elastischen Cluster wurde auf 128 TiB erhöht.
- Amazon DocumentDB 5.0 unterstützt jetzt Indexscan mit dem -`$elemMatch`Operator in der ersten Verschachtelungsebene. Index-Scans werden unterstützt, wenn die Abfrage nur eine `$elemMatch` Filterebene hat und die verschachtelte `$elemMatch` Abfrage keinen Index-Scan unterstützt.

  Abfrageform, die Indexscan unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

Abfrageform, die keinen Index-Scan unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e":
 "abc"} }} } })
```

27. Februar 2023

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 4.0

Unterstützung für hinzugefügt AWS Lambda. Weitere Informationen finden Sie unter Verwenden von AWS Lambda mit Änderungsstreams.

2. Februar 2023

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208432)

- Es wurde ein Problem mit der DML-Prüfung behoben, wenn sie während eines laufenden Workloads aktiviert ist.
- Es wurde ein Problem mit der DML-Prüfung behoben, wenn Aggregatbefehle mit Hinweis einen Zeichenfolgenwert übergeben.
- Es wurde ein Problem behoben, bei dem `listCollections` der Befehl nicht funktionierte, wenn Benutzer mit readwriteanydatabase-Rolle sowohl authorizedCollections als auch nameOnly-Optionen auf „true“ gesetzt waren.
- Es wurde ein Problem behoben, bei dem numerische Zeichenfolgen in einem Feldnamen korrekt analysiert wurden.
- Stornieren Sie lange laufende Cursor, wenn sie sich auf die Garbage Collection auswirken.

30. November 2022

Neue Features

Amazon DocumentDB-Elastic-Cluster

Elastische Amazon DocumentDB-Cluster ist ein neuer Typ von Amazon DocumentDB-Cluster, mit dem Benutzer die MongoDB-Sharding-APIs nutzen können, um ihren Cluster zu skalieren. Elastic Cluster verarbeiten praktisch jede Anzahl von Lese- und Schreibvorgängen mit einer Speicherkapazität von Petabyte, indem sie die Daten und die Datenverarbeitung auf mehrere zugrunde liegende Rechen-Instances und Volumes verteilen. Weitere Informationen finden Sie unter Verwenden von elastischen Amazon DocumentDB-Clustern.

09. August 2022

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208152) und 4.0

- Unterstützung für den Datentyp Decimal128 hinzugefügt. Decimal128 ist ein BSON-Datentyp, der in allen Regionen unterstützt wird, in denen DocumentDB verfügbar ist.

  Weitere Informationen finden Sie unter Data Types.
- Unterstützung für die Prüfung von DML-Abfragen mit Amazon CloudWatch Logs hinzugefügt. Jetzt kann Amazon DocumentDB Data Manipulation Language (DML)-Ereignisse und Data Definition Language (DDL)-Ereignisse in Amazon CloudWatch Logs aufzeichnen.

  Weitere Informationen finden Sie in diesem Blogbeitrag.

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208152) und 4.0

- Sie können jetzt Ihre eigene passwoprd mit einem eigenen Passwort mit `-changeOwnPassword`Berechtigung ändern.

25. Juli 2022

Neue Features

Amazon DocumentDB 4.0

Sie können jetzt Cluster schneller erstellen, mit der Möglichkeit, Klone zu erstellen, die dasselbe DocumentDB-Cluster-Volume verwenden und dieselben Daten wie der ursprüngliche Cluster haben. Weitere Informationen finden Sie unter Verwalten von Amazon DocumentDB-Clustern.

27. Juni 2022

Neue Features

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.7509)

Amazon DocumentDB passt die Größe Ihrer Datenbank dynamisch basierend auf Nutzungsmustern an. Das Hinzufügen weiterer Daten erhöht den Speicherplatz auf bis zu 64 Tebibyte (TiB) und das Löschen von Daten verringert den zugewiesenen Speicherplatz.

29. April 2022

Neue Features

Amazon DocumentDB ist jetzt in der Region China (Peking) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

7. April 2022

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Versionen 1.0.207836 und 1.0.208015) und 4.0 (Engine-Patch-Versionen 2.0.6142 und 2.0.6948)

Amazon DocumentDB Performance Insights befindet sich jetzt in der Vorschau. Sie können jetzt sieben Tage Leistungsverlauf ohne zusätzliche Kosten in einem fortlaufenden Fenster speichern. Weitere Informationen finden Sie unter Überwachung mit Performance Insights.

16. März 2022

Neue Features

Amazon DocumentDB ist jetzt in der Region Europa (Mailand) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

8. Februar 2022

Neue Features

Amazon DocumentDB-R6g- und T4g-Instances sind jetzt in Asien-Pazifik, Südamerika und Europa verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

24. Januar 2022

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207684) und 4.0 (Engine-Patch-Version 2.0.5170)

- DocDB ; bietet jetzt eine kostenlose Testversion an. Weitere Informationen finden Sie auf der kostenlosen Testseite von Amazon DocumentDB.
- Sie können jetzt erweiterte Funktionen mit Geospatial Query verwenden, einschließlich der folgenden APIs:
  - `$geoWithin`
  - `$geoIntersects`
- Unterstützung für die folgenden MongoDB-Operatoren hinzugefügt:
  - `$mergeObjects`
  - `$reduce`

  Weitere Informationen finden Sie unter Abfragen von Geodaten mit Amazon DocumentDB.

21. Januar 2022

Neue Features

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.5706)

- Amazon DocumentDB Graviton2 (r6g.large, r6g.2xlarge, r6g.4xlarge, r6g.8xlarge, r6g.12xlarge, r6g.16xlarge und t4g.medium)-Instances werden jetzt unterstützt

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207781) und 4.0 (Engine-Patch-Version 2.0.5706)

- Unterstützung für die folgenden MongoDB-APIs hinzugefügt:
  - `$reduce`
  - `$mergeObjects`
  - `$geoWithin`
  - `$geoIntersects`

25. Oktober 2021

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207780) und 4.0 (Engine-Patch-Version 2.0.5704)

- Unterstützung für die folgenden MongoDB-APIs hinzugefügt
  - `$literal`
  - `$map`
  - `$$ROOT`
- Unterstützung für GeoSpatial Abfragefunktionen. Weitere Informationen finden Sie in diesem Blog-Beitrag
- Unterstützung für die Zugriffskontrolle mit benutzerdefinierten Rollen. Weitere Informationen finden Sie in diesem Blog-Beitrag
- Amazon DocumentDB-JDBC-Treiber, um Konnektivität von BI-Tools wie Tableau und Abfragetools wie SQL Workbench zu ermöglichen

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207780) und 4.0 (Engine-Patch-Version 2.0.5704)

- Fehlerbehebung für `$natural` die korrekte Sortierung, wenn ein explizites zusammen mit vorhanden `.sort()` ist `$natural`
- Fehlerbehebung für den Änderungsstream, mit dem er arbeiten kann `$redact`
- Fehlerbehebung für `$ifNull`, um mit leerem Array zu arbeiten
- Fehlerbehebung bei übermäßigem Ressourcenverbrauch/Serverabsturz, wenn ein aktuell angemeldeter Benutzer gelöscht wird oder die Berechtigung dieses Benutzers für eine laufende Aktivität widerrufen wird
- Fehlerbehebung in `listDatabase` und `listCollection` Berechtigungsprüfung
- Fehlerbehebung bei der Deduplizierungslogik für Elemente mit mehreren Schlüsseln

24. Juni 2021

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207117) und 4.0 (Engine-Patch-Version 2.0.3371)

- r5.8xlarge- und r5.16xlarge-Instances werden jetzt unterstützt. Weitere Informationen finden Sie im Blogbeitrag Amazon DocumentDB unterstützt jetzt r5.8xlarge- und r5.16xlarge-Instances.
- Globale Cluster werden jetzt unterstützt, um eine Notfallwiederherstellung nach regionsweiten Ausfällen zu ermöglichen und globale Lesevorgänge mit niedriger Latenz zu ermöglichen, indem Lesevorgänge aus dem nächstgelegenen Amazon DocumentDB-Cluster zugelassen werden.

4. Mai 2021

Neue Features

Sehen Sie sich alle neuen Funktionen in diesem Blogbeitrag an.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207117) und 4.0 (Engine-Patch-Version 2.0.3371)

- `renameCollection`
- `$zip`
- `$indexOfArray`
- `$reverseArray`
- `$natural`
- `$hint` -Unterstützung für -Update
- Index-Scan für `distinct`

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207117) und 4.0 (Engine-Patch-Version 2.0.3371)

- Geringere Speicherauslastung für `$in` Abfragen
- Es wurde ein Speicherleck in Multikey-Indizes behoben.
- Die Ausgabe des Erläuterungsplans und des Profilers für wurde korrigiert. `$out`
- Es wurde ein Timeout für Operationen aus dem internen Überwachungssystem hinzugefügt, um die Zuverlässigkeit zu verbessern.
- Es wurde ein Fehler behoben, der sich auf die Abfrageprädikate auswirkte, die an Multischlüssel-Indizes übergeben wurden.

15. Januar 2021

Neue Features

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.722)

- None

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Möglichkeit, einen Index mit der `$lookup` Aggregationsphase zu verwenden
- `find()` -Abfragen mit Projektionen können von einem Index aus in die Richtung bedient werden (abgedeckte Abfrage)

- Möglichkeit zur Verwendung `hint()` mit der `findAndModify`
- Leistungsoptimierungen für `$addToSet` Operator
- Verbesserungen zur Reduzierung der Gesamtindexgrößen
- Neue Aggregationsoperatoren: `$ifNull`, `$setInstersection$replaceRoot`, `$setIsSubset`, `$setUnion`, und `$setEquals`
- Benutzer können auch ihre eigenen Cursor beenden, ohne die `KillCursor` Rolle zu benötigen

9. November 2020

Neue Features

Sehen Sie sich alle neuen Funktionen in diesem Blogbeitrag an.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.722)

- Kompatibilität mit MongoDB 4.0
- ACID-Transaktionen
- Unterstützung für `cluster(client.watch()` oder `mongo.watch())` und die `(db.watch())` Änderungsdatenströme auf Datenbankebene
- Möglichkeit zum Starten oder Fortsetzen eines Änderungsstreams mit `startAtOperationTime`
- Verlängern Sie den Aufbewahrungszeitraum für Änderungsstreams auf 7 Tage (zuvor 24 Stunden)
- AWS DMS -Ziel für Amazon DocumentDB 4.0
- CloudWatch -Metriken: `TransactionsOpen`, `TransactionsOpenMaxTransactionsAborted`, `TransactionsStarted`, und `TransactionsCommitted`
- Neue Felder für Transaktionen in `currentOp`, `ServerStatus`und `profiler`.
- Möglichkeit, einen Index mit der `$lookup` Aggregationsphase zu verwenden
- `find()` -Abfragen mit Projektionen können von einem Index aus in die Richtung bedient werden (abgedeckte Abfrage)
- Möglichkeit zur Verwendung `hint()` mit der `findAndModify`
- Leistungsoptimierungen für `$addToSet` Operator
- Verbesserungen zur Reduzierung der Gesamtindexgrößen.
- Neue Aggregationsoperatoren: `$ifNull`, `$setInstersection$replaceRoot`, `$setIsSubset`, `$setUnion`, und `$setEquals`

- Mit den `ListDatabase` Befehlen `ListCollection` und können Sie jetzt optional die `authorizedDatabases` Parameter `authorizedCollections` und verwenden, um Benutzern das Auflisten der Sammlungen und Datenbanken zu ermöglichen, auf die sie zugreifen können, ohne dass die `listDatabase` Rollen `listCollections` und erforderlich sind.
- Benutzer können auch ihre eigenen Cursor beenden, ohne die `KillCursor` Rolle zu benötigen
- Der Vergleich numerischer Unterdokumenttypen ist jetzt konsistent mit dem Vergleich numerischer Dokumenttypen der ersten Ebene. Das Verhalten in Amazon DocumentDB 4.0 ist jetzt mit MongoDB kompatibel.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- None

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.722)

- `$setOnInsert` lässt keine Aktualisierungen mehr zu, wenn der Positionsoperator verwendet wird$. Das Verhalten in Amazon DocumentDB 4.0 ist jetzt mit MongoDB kompatibel.
- Es wurde ein Problem mit `$createCollection` und Set behoben. `autoIndexId`
- Projektion für verschachtelte Dokumente
- Die Standardeinstellung für den Arbeitsspeicher zur Skalierung mit der Instance-Speichergröße wurde geändert
- Verbesserungen der Garbage Collection
- Suche mit leerem Schlüssel im Pfad, Verhaltensunterschied mit Mongo
- Es wurde `dateToString` ein Fehler im Zeitzonenverhalten behoben.
- Behebung `$push` (Aggregation), um die Sortierreihenfolge zu berücksichtigen
- Es wurde ein Fehler in `$currentOp` mit Aggregat behoben.
- Es wurde ein Problem mit `readPreference` auf sekundärem behoben.
- Es `$createIndex` wurde ein Problem mit der Validierung behoben, bei dem es sich um dieselbe Datenbank handelt, in der der Befehl ausgegeben wurde.
- Inkonsistentes Verhalten für behoben`minKey`, `maxKey` Suche schlägt fehl
- Es wurde ein Problem behoben, bei dem der `$size` Operator nicht mit dem zusammengesetzten Array arbeitete.

- Es wurde ein Problem mit der Negation von `$in` mit Regex behoben.
- Es wurde ein Problem mit `$distinct` der Befehlsausführung für eine Ansicht behoben.
- Es wurde ein Problem mit Aggregationen behoben und Befehle zum unterschiedlichen Sortieren fehlender Felder gefunden.
- Es wurde behoben`$eq`, dass reguläre Ausdrücke den Typ nicht überprüfen.
- Es wurde `$currentDate` ein Fehler im Verhalten der ordinalen Position des Zeitstempels behoben.
- Granularität in Millisekunden für behoben `$currentDate`

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- None

30. Oktober 2020

Neue Features

Sehen Sie sich alle neuen Funktionen in diesem Blogbeitrag an.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Möglichkeit hinzugefügt, einen Änderungsstream-Cursor auf Cluster-Ebene (`client.watch()` oder `mongo.watch()`) und der Datenbank zu öffnen (`db.watch()`)
- Möglichkeit, den Aufbewahrungszeitraum für Änderungsstreams auf 7 Tage zu erhöhen (zuvor 24 Stunden)

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Verschiedene allgemeine Verbesserungen der Fallleistung
- Eine gezielte Sicherheitsverbesserung
- Es wurde ein Problem mit dem Überspringen der Sortierung im zweiten Feld eines zusammengesetzten Indexes behoben.

- Aktivieren Sie den regulären Index für die Gleichheit auf einem einzelnen Feld eines Multi-Schlüssel-Index (nicht zusammengesetzt)
- Authentifizierungs-Race-Bedingung behoben
- Es wurde ein Problem behoben, das zu einem seltenen Absturz der Garbage Collection führte.
- Verbesserung der RBAC-Sicherheit
- `databaseConnectionsMax` Metrik hinzugefügt
- Leistungsverbesserungen für bestimmte Workloads auf `r5.24xlarge`-Instances

22. September 2020

Neue Features

Sehen Sie sich alle neuen Funktionen in diesem Blogbeitrag an.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- `$out` Aggregationsphase
- Die maximale Anzahl von Verbindungen und Cursor pro Instance wurde um das 10-fache erhöht

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- None

10. Juli 2020

Neue Features

Sehen Sie sich alle neuen Funktionen in diesem Blogbeitrag an.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Regionsübergreifende Snapshot-Kopie

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- None

30. Juni 2020

Neue Features

Sehen Sie sich alle neuen Funktionen in diesem Blogbeitrag an.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- T3-medium-Instances

Fehlerbehebungen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Wiederherstellung des inaktiven Speichers für t3-Instances
- Verbesserungen der Authentifizierung
- Verbesserte SASL-Authentifizierungsleistung
- Es wurde ein `currentOp` Problem behoben, bei dem die maximal möglichen Operationen überschritten wurden.
- Es wurde ein `killOps` Problem mit Massenaktualisierungen und -löschungen behoben.
- Verbesserungen der `$sample` Leistung mit `$match`
- Unterstützung für `$$` in Cond-Case in der Redigierungsphase korrigiert
- Es wurden verschiedene Ursachen für wiederkehrende Abstürze behoben.
- Verbesserungen beim TTL-Sweeping zur Reduzierung IOs und Latenz
- Optimierte Speicherauslastung für `$unwind`
- Race-Bedingung für Sammlungsstatistiken mit Drop-Index behoben
- Race-Bedingung beim gleichzeitigen Indexaufbau behoben
- Es wurde ein seltener Absturz in `hash_search` im Index behoben.

Dokumentverlauf für das Amazon DocumentDB-Entwicklerhandbuch

- API-Version: 2014-10-31
- Letzte Aktualisierung der Dokumentation: 2. Juni 2023

In der folgenden Tabelle wird die Dokumentation für diese Version des Amazon DocumentDB-Entwicklerhandbuchs beschrieben.

Änderung	Beschreibung	Datum
AWS -verwaltete Richtlinienaktualisierung – Richtlinienänderung	Amazon DocumentDB aktualisiert die Richtlinien für vollständigen Zugriff für Elastic Cluster.	21. Februar 2024
AWS -verwaltete Richtlinienaktualisierung – Richtlinienänderung	Amazon DocumentDB aktualisiert die Richtlinien für Lese- und Vollzugriff für elastische Cluster.	21. Juni 2023
AWS Aktualisierung der von verwalteten Richtlinie – neue Richtlinie	Amazon DocumentDB führt eine neue schreibgeschützte Richtlinie für Elastic Cluster ein.	08. Juni 2023
AWS Aktualisierung der von verwalteten Richtlinie – neue Richtlinie	Amazon DocumentDB führt eine neue Vollzugriffsrichtlinie für Elastic Cluster ein.	5. Juni 2023
Kompatibilität mit MongoDB 5.0	Amazon DocumentDB ist jetzt mit Version 5.0 von MongoDB kompatibel.	1. März 2023
Richtlinienaktualisierung	Um die Funktion für elastische Amazon DocumentDB-	30. November 2022

	Cluster zu unterstützen, wird die AmazonDocDB-Consol eFullAccess Richtlinie aktualisi ert und die AmazonDocDB-ElasticServiceRolePolicy wird eingeführt.	
Elastic Cluster	Es wurde eine neue Elastic-Cluster-Funktion hinzugefügt, die die Hash-basierte Partition ierung (Sharding) von Daten im verteilten Speichersystem von Amazon DocumentDB unterstützt.	30. November 2022
Globale Cluster	Dokumentation zur Verwendun g von globalen Clustern hinzugefügt.	2. Juni 2021
Ereignisabonnements	Dokumentation zum Ereignisa bonnement hinzugefügt.	26. März 2021
Version 3.6 Upgrades	Dokumentierte Verbesser ungen an Version 3.6 in rollenbasierten Zugriffsk ontrollen, Aggregationsoperat oren und Leistung.	15. Januar 2021
Kompatibilität mit MongoDB 4.0	Amazon DocumentDB ist jetzt mit Version 4.0 von MongoDB kompatibel.	9. November 2020
Erste Schritte	Neue Anleitungen für die ersten Schritte mit Amazon DocumentDB unter Verwendung von AWS Cloud9, Amazon EC2, Robo3T oder Studio3T.	15. August 2020

Zusätzliche unterstützte Availability Zones	Amazon DocumentDB hat Unterstützung für eine zusätzliche Availability Zone in Asien-Pazifik (Seoul) (ap-northeast-2) hinzugefügt.	14. Juli 2020
Unterstützung für das regionsübergreifende Kopieren von Snapshots hinzugefügt.	Amazon DocumentDB hat Unterstützung für das Kopieren von Cluster-Snapshots über hinweg hinzugefügt AWS-Regionen. Weitere Informationen finden Sie unter Kopieren von Snapshots über Regionen hinweg.	10. Juli 2020
Unterstützung für die T3-Instance-Klasse hinzugefügt.	Unterstützung für T3-Instance-Typen in allen Regionen hinzugefügt, die Amazon DocumentDB unterstützen. Weitere Informationen finden Sie unter Unterstützte Instance-Klassen nach Regionen und Instance-Klassenspezifikationen.	30. Juni 2020
Unterstützung für hinzugefügt AWS GovCloud (US).	Amazon DocumentDB ist jetzt in der AWS GovCloud (US) Region (us-gov-west-1) verfügbar.	29. Juni 2020

Es wurden 16 neue CloudWatch Metriken hinzugefügt.	Amazon DocumentDB hat Unterstützung für 16 neue Amazon- CloudWatch Metriken hinzugefügt. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.	23. Juni 2020
Unterstützung für Nullzeichen und $regex-Operator hinzugefügt.	Amazon DocumentDB hat Unterstützung für Nullzeichen in Zeichenfolgen und die Möglichkeit hinzugefügt, einen Index für $regex zu verwenden. Informationen zu den unterstützten MongoDB-APIs und Aggregations-Pipeline-Funktionen für Amazon DocumentDB finden Sie unter Funktionsunterschiede zu MongoDB.	22. Juni 2020
Unterstützung für verbesserte Funktionen zur Indizierung mehrerer Schlüssel hinzugefügt.	Amazon DocumentDB hat Unterstützung für verbesserte Funktionen zur Indizierung mehrerer Schlüssel hinzugefügt, die die Indizierung von Arrays mit mehr als 2 048 Byte und die Möglichkeit umfassen, einen zusammengesetzten Multi-Schlüssel-Index mit mehreren Schlüsseln im selben Array zu erstellen. Weitere Informationen finden Sie unter Funktionsunterschiede bei MongoDB.	23. April 2020

Unterstützung für den Löschschutz für einen Amazon DocumentDB AWS CloudFormation -Stack hinzugefügt.	Amazon DocumentDB hat Unterstützung für die Aktivierung des Löschschutzes beim Erstellen eines Amazon DocumentDB AWS CloudFormation -Stacks hinzugefügt.	20. April 2020
Unterstützung für die rollenbasierte Zugriffskontrolle hinzugefügt.	Amazon DocumentDB hat mithilfe integrierter Rollen Unterstützung für die rollenbasierte Zugriffskontrolle hinzugefügt.	26. März 2020
Unterstützung für eine zusätzliche Availability Zone in Kanada (Zentral) (ca-central-1) hinzugefügt.	Amazon DocumentDB ist jetzt in der Region Kanada (Zentral) (ca-central-1) mit R5-Klassen-Instances und 3 Availability Zones verfügbar.	26. März 2020
Unterstützung für zwei zusätzliche MongoDB-APIs hinzugefügt.	Amazon DocumentDB hat Unterstützung für `$dateFromString` und `executionStats` MongoDB-APIs hinzugefügt.	23. März 2020
Unterstützung für fünf zusätzliche MongoDB-APIs hinzugefügt.	Amazon DocumentDB hat Unterstützung für die APIs `$objectToArray` , `$slice$arrayTo0bject` , `$mod`, und `$range` MongoDB hinzugefügt APIs.	6. Februar 2020
Unterstützung für Kanada (Zentral) hinzugefügt.	Amazon DocumentDB ist jetzt in der Region Kanada (Zentral) (ca-central-1) mit R5-Klassen-Instances verfügbar.	11. Dezember 2019

Unterstützung für hinzugefügt ChangeStreamLogSize.	Amazon DocumentDB hat Unterstützung für `ChangeStreamLogSize` für Cloudwatch-Metriken hinzugefügt.	22. November 2019
Unterstützung für die Region Europa (Paris) hinzugefügt	Amazon DocumentDB ist jetzt in der Region Europa (Paris) (eu-west-3) mit R5-Klassen-Instances verfügbar.	30. Oktober 2019
Unterstützung für die Region Asien-Pazifik (Mumbai) hinzugefügt	Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Mumbai) (ap-south-1) mit R5-Klassen-Instances verfügbar.	17. Oktober 2019
Unterstützung für drei zusätzliche MongoDB-APIs hinzugefügt	Amazon DocumentDB hat Unterstützung für die `$concatArrays` -, `$addFields` - und `$lookup` MongoDB-APIs hinzugefügt.	16. Oktober 2019
Unterstützung für die Region Asien-Pazifik (Singapur) hinzugefügt	Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Singapur) (ap-southeast-1) mit R5-Klassen-Instances verfügbar.	14. Oktober 2019
Neues Dokument zum Aktualisieren von TLS-Zertifikaten hinzugefügt	Anweisungen zum Aktualisieren von Zertifizierungsstellenzertifikaten zur Verwendung des neuen Zertifizierungsstellenzertifikats zum Erstellen von TLS-Verbindungen wurden hinzugefügt.	2. Oktober 2019

API-Unterstützung für Zertifikate hinzugefügt	Amazon DocumentDB ist ein neuer Zertifikat-Datentyp für Instances. Weitere Informationen finden Sie unter DBInstance.	1. Oktober 2019
Unterstützung für Abfrageprofilerstellung	Amazon DocumentDB hat die Möglichkeit hinzugefügt, unterstützte Operationen auf den Instances und Datenbanken Ihres Clusters profilieren zu können.	19. August 2019
Die dritte AZ wurde in Asien-Pazifik (Tokio) hinzugefügt	Amazon DocumentDB hat eine dritte Availability Zone (AZ) für Ihre Datenverarbeitungs-Instances in Asien-Pazifik (Tokio) hinzugefügt.	9. August 2019
Unterstützung für zusätzliche Mongo-APIs	Unterstützung für zusätzliche Aggregationspipeline-Funktionen hinzugefügt, zu denen die `$dateToString` Aggregationsoperatoren `$in`, `$isoWeek` `$isoWeekYear` , `$isoDayOfWeek` , und sowie die `$addToSet` Aggregationsphase gehören. Amazon DocumentDB hat auch Unterstützung für den `top()` Befehl für die Diagnose auf Sammlungsebene und die Möglichkeit hinzugefügt, den `expireAfterSeconds` Parameter für TTL-Indizes mithilfe des `collMod()` Befehls zu ändern.	31. Juli 2019

Unterstützung für Europa (London) hinzugefügt	Amazon DocumentDB ist jetzt in Europa (London) (eu-west-2) mit R5-Klassen-Instances verfügbar.	18. Juli 2019
Codebeispiele hinzugefügt	Codebeispiele in R und Ruby für die programmgesteuerte Verbindung mit Amazon DocumentDB hinzugefügt.	17. Juli 2019
Bewährte Methoden hinzugefügt	Es wurde eine bewährte Methode hinzugefügt, die Sie bei der Verwaltung Ihrer Amazon DocumentDB-Kosten unterstützt.	17. Juli 2019
Unterstützung für das Stoppen und Starten eines Clusters	Amazon DocumentDB hat Unterstützung für das Stoppen und Starten von Clustern hinzugefügt, um die Kosten für Entwicklungs- und Testumgebungen zu verwalten.	1. Juli 2019
Unterstützung für den Cluster-Löschschutz	Um Ihre Cluster vor versehentlichem Löschen zu schützen, hat Amazon DocumentDB Löschschutz hinzugefügt. Weitere Informationen finden Sie in den folgenden Themen: Erstellen eines Amazon DocumentDB-Clusters, Ändern eines Amazon DocumentDB-Clusters, Löschen eines Amazon DocumentDB-Clusters und `DeletionProtection` im API-Thema DBCluster.	1. Juli 2019

Aktualisierung der Funktionsunterschiede	Implizite Transaktionen wurden den Funktionsunterschieden hinzugefügt.	26. Juni 2019
Ergänzung zu funktionalen Unterschieden	Hinweis zur Speicher- und Indexkomprimierung in Amazon DocumentDB hinzugefügt.	13. Juni 2019
Zusätzliche unterstützte Region	Amazon DocumentDB ist jetzt in Asien-Pazifik (Sydney) (ap-southeast-2) mit R5-Klassen-Instances verfügbar.	5. Juni 2019
In weiteren Regionen unterstützte R5-Instance-Klasse	Unterstützung für R5-Instance-Klasse für 4 zusätzliche Regionen hinzugefügt: USA Ost (Ohio), USA Ost (Nord-Virginia), USA West (Oregon) und EU (Irland). Mit dieser Änderung werden R5-Instances in allen Regionen unterstützt, die Amazon DocumentDB unterstützen.	17. Mai 2019
Zusätzliche unterstützte Regionen	Unterstützung für zwei zusätzliche Regionen hinzugefügt: Asien-Pazifik (Tokio) (ap-northeast-1) und Asien-Pazifik (Seoul) (ap-northeast-2) mit R5-Instance-Klassen. Weitere Informationen finden Sie unter Unterstützte Instance-Klassen nach Region und Instance-Klassen-Spezifikationen.	8. Mai 2019

Weitere Beispiele für Verbindungscodes hinzugefügt	Codebeispiele in Java und C# für die Verbindung mit Amazon DocumentDB hinzugefügt.	24. April 2019
Zusätzliche Mongo-API-Unterstützung	Es wurde die Unterstützung von sieben Aggregations-Zeichenfolgenoperatoren (`$indexOfBytes`, `$indexOfCP`, `$strLenBytes`, `$strLenCP`, `$toLower`, `$toUpper`, `$split`), neun Datumszeitoperatoren (`$dayOfYear`, `$dayOfMonth`, `$dayOfWeek`, `$year`, `$month`, `$hour`, `$minute`, `$second`, `$millisecond`) und die Aggregations-Pipeline-Stufe `$sample` hinzugefügt.	4. April 2019
Beispiele für Verbindungscodes hinzugefügt	Codebeispiele in Python, Node.js, PHP und Go für die Verbindung mit Amazon DocumentDB hinzugefügt.	21. März 2019
Unterstützung für Region Frankfurt und R5-Instances	Unterstützung für die Region Europa (Frankfurt) (eu-central-1) mit R5-Instance-Klassen hinzugefügt. Weitere Informationen finden Sie unter Unterstützte Instance-Klassen nach Region und Instance-Klassen-Spezifikationen.	13. März 2019

Unterstützung für Aggregations-Pipeline-Operatoren	Es wurde die Unterstützung für neue Aggregations-Zeichenfolgenoperatoren (`$concat`, `$substr`, `$substrBytes` , `$substrCP` , `$strcasecmp`), einen Array-Aggregationsoperator (`$size`), einen Aggregationsgruppen-Akkumulatoroperator (`$push`) und Aggregationsstufen (`$redact` und `$indexStats`) hinzugefügt. Wir haben außerdem die Unterstützung für Positional-Array-Operatoren (`$[]` und `$[<identifier>]`) und `hint()` hinzugefügt.	28. Februar 2019
Engine-Upgrades	Es wurde eine Dokumentation zur Ermittlung ausstehender Cluster-Änderungen und Aktualisierung der Engine Version Ihres Clusters hinzugefügt.	15. Februar 2019
Prüfen von Ereignissen	Unterstützung für die Prüfung von Datenbankereignissen mit Amazon CloudWatch Logs hinzugefügt.	12. Februar 2019
Schnellstart	Es wurde ein Schnellstartthema hinzugefügt, das Ihnen hilft, einfach mit Amazon DocumentDB mit zu beginnen AWS CloudFormation.	11. Januar 2019

Öffentliche Veröffentlichung	Dies ist die erste öffentliche Veröffentlichung von Amazon DocumentDB (mit MongoDB-Kompatibilität). Diese Version enthält den Entwicklerleitfaden und die integrierte API-Referenz für das Ressourcenmanagement.	9. Januar 2019

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.