



Entwicklerhandbuch

Amazon Keyspaces (für Apache Cassandra)



Amazon Keyspaces (für Apache Cassandra): Entwicklerhandbuch

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon Keyspaces?	1
Funktionsweise	2
Hochrangige Architektur	2
Cassandra-Datenmodell	5
Zugreifen auf Amazon Keyspaces	6
Anwendungsfälle	7
Was ist CQL?	8
Vergleichen Sie Amazon Keyspaces mit Cassandra	9
Funktionelle Unterschiede zu Apache Cassandra	10
APIs, Operationen und Datentypen von Apache Cassandra	11
Asynchrones Erstellen und Löschen von Schlüsselräumen und Tabellen	11
Authentifizierung und Autorisierung	11
Stapel	11
Cluster-Konfiguration	11
Verbindungen	12
INSchlüsselwort	12
Optimierung des Durchsatzes für CQL-Abfragen	13
FROZENSammlungen	14
Einfache Transaktionen	14
Load Balancing	14
Paginierung	15
Partitionierer	15
Vorbereitete Aussagen	15
Bereich löschen	15
Systemtabellen	16
Zeitstempel	16
Unterstützte Cassandra-APIs, Operationen, Funktionen und Datentypen	16
Cassandra-API-Unterstützung	17
API-Unterstützung für die Cassandra-Steuerebene	18
API-Unterstützung für Cassandra-Datenebene	19
Unterstützung der Cassandra-Funktion	19
Unterstützung des Cassandra-Datentyps	20
Unterstützte Cassandra-Konsistenzstufen	22
Konsistenzstufen schreiben	22

Konsistenzstufen lesen	23
Nicht unterstützte Konsistenzstufen	24
Zugreifen auf Amazon Keyspaces	25
Einrichten AWS Identity and Access Management	25
Melden Sie sich an für ein AWS-Konto	25
Erstellen Sie einen Benutzer mit Administratorzugriff	25
Amazon Keyspaces einrichten	27
Verwenden der Konsole	28
Benutzen AWS CloudShell	28
Erhalt von IAM-Berechtigungen für AWS CloudShell	29
Interaktion mit Amazon Keyspaces über AWS CloudShell	30
Programmgesteuert eine Verbindung herstellen	32
Anmeldeinformationen erstellen	33
Service-Endpunkte	45
Verwenden von <code>cqlsh</code>	49
Verwendung der AWS CLI	57
Verwenden der API	62
Mit SDKs AWS arbeiten	62
Verwenden eines Cassandra-Client-Treibers	63
Herstellen einer Verbindung von Amazon EKS	95
Verbindung mit VPC-Endpunkten herstellen	116
Voraussetzungen	117
Schritt 1: Starten einer Amazon-EC2-Instance	117
Schritt 2: Konfigurieren Ihrer Amazon-EC2-Instance	119
Schritt 3: Erstellen Sie einen VPC-Endpunkt für Amazon Keyspaces	122
Schritt 4: Berechtigungen für die VPC-Endpunktverbindung konfigurieren	127
Schritt 5: Konfigurieren Sie die Überwachung	131
Schritt 6: (Optional) Bewährte Methoden für Verbindungen	132
Schritt 7: (Optional) Aufräumen	135
Kontenübergreifender Zugriff	136
Kontoübergreifende Zugriff	136
Kontoübergreifende Zugriff	140
Erste Schritte	142
Voraussetzungen	142
Schritt 1: Erstellen Sie einen Schlüsselraum und eine Tabelle	143
Einen Schlüsselraum erstellen	143

Erstellen einer Tabelle	145
Schritt 2: CRUD-Operationen	150
Erstellen	150
Lesen	152
Aktualisierung	155
Löschen	156
Schritt 3: Aufräumen (optional)	157
Löschen einer Tabelle	157
Einen Schlüsselraum löschen	159
Migration zu Amazon Keyspaces	161
Migration von Cassandra	163
Kompatibilität	163
Schätzen Sie den Preis	164
Migrationsstrategie	173
Offline-Migration	174
Migrationstools	176
Laden von Daten mit cqlsh	177
Daten mit DSBulk laden	189
Codebeispiele	202
Aktionen	207
CreateKeyspace	208
CreateTable	212
DeleteKeyspace	218
DeleteTable	222
GetKeyspace	226
GetTable	229
ListKeyspaces	234
ListTables	238
RestoreTable	242
UpdateTable	246
Szenarien	251
Beginnen Sie mit Schlüsselräumen und Tabellen	251
Bibliotheken und Tools	315
Bibliotheken und Beispiele	315
Amazon Keyspaces (für Apache Cassandra) Entwickler-Toolkit	315
Beispiele für Amazon Keyspaces (für Apache Cassandra)	315

AWSSignature Version 4 (SigV4) Authentifizierungs-Plugins	315
Hervorgehobene Repos für Beispiel- und Entwicklertools	316
Amazon Keyspaces-Protokollpuffer	316
AWS CloudFormationVorlage zur Erstellung eines CloudWatch Amazon-Dashboards für Amazon Keyspaces-Metriken (für Apache Cassandra)	316
Verwenden von Amazon Keyspaces (für Apache Cassandra) mit AWS Lambda	317
Amazon Keyspaces (für Apache Cassandra) mit Spring verwenden	317
Amazon Keyspaces (für Apache Cassandra) mit Scala verwenden	317
Verwenden von Amazon Keyspaces (für Apache Cassandra) mit AWS Glue	317
Amazon Keyspaces (für Apache Cassandra) Cassandra-Abfragesprache (CQL) zum Konverter AWS CloudFormation	318
Amazon Keyspaces (für Apache Cassandra) Helfer für den Apache Cassandra-Treiber für Java	318
Schnelle Kompressionsdemo von Amazon Keyspaces (für Apache Cassandra)	318
Amazon Keyspaces (für Apache Cassandra) und Amazon S3-Codec-Demo	318
Integration mit Apache Spark	319
Voraussetzungen	320
Schritt 1: Amazon Keyspaces konfigurieren	320
Schritt 2: Den Apache Cassandra Spark Connector konfigurieren	322
Schritt 3: Erstellen Sie die App-Konfigurationsdatei	324
Stellen Sie eine Verbindung mit der SIGv4-Authentifizierung her	324
Stellen Sie eine Verbindung mit dienstspezifischen Anmeldeinformationen her	325
Stellen Sie eine Verbindung zu einem festen Tarif her	326
Schritt 4: Bereiten Sie die Quelldaten und die Zieltabelle vor	327
Schritt 5: Amazon Keyspaces-Daten schreiben und lesen	329
Fehlerbehebung	332
Häufige Fehler und Warnungen	333
Fehlerbehebung	334
Allgemeine Fehler	334
Allgemeine Fehler	334
Verbindungen	336
Fehler beim Herstellen einer Verbindung zu einem Amazon Keyspaces-Endpunkt	336
Kapazitätsverwaltung	349
Kapazitätsfehler bei serverlosen Servern	349
Sprache der Datendefinition	355
Sprachfehler bei der Datendefinition	355

Serverloses Ressourcenmanagement	361
Speicher	361
Lese-/Schreibkapazitätsmodi	362
On-Demand-Kapazitätsmodus	362
Bereitgestellter Durchsatzkapazitätsmodus	365
Kapazitätsmodi verwalten und anzeigen	367
Überlegungen beim Ändern der Kapazitätsmodi	368
Verwaltung der Durchsatzkapazität mit Auto Scaling	369
So funktioniert die automatische Skalierung von Amazon Keyspace	369
So funktioniert Auto Scaling für Tabellen mit mehreren Regionen	371
Nutzungshinweise	372
Verwenden der Konsole	373
Verwenden von CQL	378
Verwenden der -CLI	384
Burst-Kapazität	390
Schätzen Sie den Kapazitätsverbrauch	391
Bereichsabfragen	392
Abfragen einschränken	392
Tabellenscans	393
Leichte Transaktionen	394
Schätzen Sie den Lese- und Schreibkapazitätsverbrauch mit Amazon CloudWatch	394
Arbeiten mit Amazon Keyspaces	396
Arbeiten mit Keyspaces	396
System-Keyspaces	396
Erstellen von Keyspaces	403
Arbeiten mit Tabellen	404
Erstellen von Tabellen	404
Multiregionale Tabellen	405
Statische Spalten	407
Mit Zeilen arbeiten	411
Berechnung der Zeilengröße	411
Arbeiten mit Abfragen	415
INSELECTAussage	415
Ergebnisse bestellen	419
Ergebnisse paginieren	421
Mit Partitionierern arbeiten	421

Arbeiten mit Tags	424
Tagging-Einschränkungen	425
von Operationen	425
Kostenzuordnungsberichte für Amazon Keyspaces	430
Bewährte Methoden	432
NoSQL-Design	433
NoSQL gegenüber RDBMS	434
Zwei Schlüsselkonzepte	434
Allgemeiner Ansatz	435
Verbindungen	436
Wie funktionieren sie	437
Wie konfiguriert man Verbindungen	438
VPC-Endpunktverbindungen	439
Wie überwacht man Verbindungen	440
Wie gehe ich mit Verbindungsfehlern um	442
Datenmodellierung	442
Entwerfen von Partitionsschlüsseln	443
Kostenoptimierung	445
Auswerten Ihrer Kosten auf Tabellenebene	446
Bewerten Sie den Kapazitätsmodus Ihrer Tabelle	448
Evaluieren Sie die Application Auto Scaling Scaling-Einstellungen Ihrer Tabelle	453
Identifizieren Sie Ihre ungenutzten Ressourcen	460
Auswerten Ihrer Tabellennutzungsmuster	466
Auswerten Ihrer bereitgestellten Kapazität, um eine angemessene Bereitstellung zu erzielen	467
NoSQL-Workbench	477
Herunterladen	478
Erste Schritte	478
Data Modeler	480
Erstellen eines Datenmodells	481
Bearbeiten eines Datenmodells	483
Data Visualizer	485
Visualisieren eines Datenmodells	485
Aggregierte Ansicht	487
Übergeben eines Datenmodells	489
Bevor Sie beginnen	490

Verbindung mit dienstspezifischen Anmeldeinformationen herstellen	491
Verbindung mit IAM-Anmeldeinformationen herstellen	494
Verwenden einer gespeicherten Verbindung	497
Apache Cassandra	497
Beispieldatenmodelle	500
Mitarbeiterdatenmodell	500
Datenmodell für Kreditkartentransaktionen	500
Datenmodell für den Flugbetrieb	501
Versionsverlauf	501
Multiregionale Replikation	503
Vorteile	503
Kapazitätsmodi und Preise	505
Funktionsweise	505
Funktionsweise	506
Konfliktlösung	507
Notfallwiederherstellung	508
IAM-Berechtigungen	509
Integration mit PITR	510
Integration mit Diensten AWS	510
Nutzungshinweise	510
So verwenden Sie die Multi-Region-Replikation	512
Verwenden der Konsole	513
Verwenden von CQL	520
Verwenden der AWS CLI	528
Point-in-time P-Wiederherstellung	538
Funktionsweise	538
Aktivieren von PITR	539
Berechtigungen wiederherstellen	542
Kontinuierliche Backups	545
Einstellungen wiederherstellen	546
PITR- und verschlüsselte Tabellen	547
PITR- und multiregionale Tabellen	548
Tabellenwiederherstellungszeit	548
Integration in AWS-Services	510
Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt	550
Bevor Sie beginnen	550

Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt (Konsole)	550
Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit demAWS CLI	552
Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit CQL	555
Wiederherstellen einer gelöschten Tabelle mit demAWS CLI	557
Wiederherstellung einer gelöschten Tabelle mit CQL	558
Ablaufende ende Daten mit Time to Live	560
Funktionsweise	561
TTL-Standardwert	562
Benutzerdefinierte TTL-Werte	562
TTL aktivieren	563
Integration in AWS-Services	563
Verwenden von Time to Live to Live	564
Erstellen einer neuen Tabelle mit aktivierten TTL-Einstellungen (Konsole)	564
Um die TTL-Einstellungen (für Gültigkeitsdauer)	565
Deaktivieren von für Gültigkeitsdauer to Live)	565
Erstellen einer neuen Tabelle mit den TTL-Einstellungen (für Gültigkeitsdauer)	566
Verwenden ALTER TABLE von für Gültigkeitsdauer to Live)	566
Aktivieren von für Gültigkeitsdauer (TTL)	567
Aktivieren von für Gültigkeitsdauer (TTL)	567
Verwenden INSERT von für Gültigkeitsdauer to Live)	567
Verwenden UPDATE von für Gültigkeitsdauer to Live)	568
ClientZeitstempel	570
Funktionsweise	571
Clientseitige Zeitstempel in Amazon Keyspaces	571
Integration in AWS-Services	572
Wie man die Zeitstempel in Betracht ziehen kann	572
Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln (Konsole)	573
Clientseitige Zeitstempel für bestehende Tabellen aktivieren (Konsole)	573
Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln (CQL)	574
Clientseitige Zeitstempel für bestehende Tabellen mithilfe vonALTER TABLE (CQL) aktivieren	574
Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln (CLI)	575
Clientseitige Zeitstempel für eine vorhandene Tabelle aktivieren (CLI)	577
Verwendung von clientseitigen Zeitstempeln in DML-Anweisungen (Data Manipulation Language)	578
AWS CloudFormation-Ressourcen	580

Amazon Keyspaces und AWS CloudFormation Vorlagen	580
Weitere Informationen zu AWS CloudFormation	580
Überwachung von Amazon Keyspaces	582
Überwachung mit CloudWatch	583
Verwenden von -Metriken	584
Metriken und Dimensionen	586
Erstellen von Alarmen	607
Protokollierung mit CloudTrail	608
Konfiguration von Protokolldateieinträgen in CloudTrail	609
DDL-Informationen in CloudTrail	610
DML-Informationen in CloudTrail	610
Grundlagen zu -Protokolldateieinträgen	611
Sicherheit	623
Datenschutz	624
Verschlüsselung im Ruhezustand	625
Verschlüsselung während der Übertragung	647
Richtlinie für den Datenverkehr zwischen Netzwerken	648
AWS Identity and Access Management	649
Zielgruppe	650
Authentifizierung mit Identitäten	651
Verwalten des Zugriffs mit Richtlinien	654
So funktioniert Amazon Keyspaces mit IAM	657
Beispiele für identitätsbasierte Richtlinien	663
Von AWS verwaltete Richtlinien	670
Fehlerbehebung	679
Verwenden von serviceverknüpften Rollen	683
Compliance-Validierung	690
Ausfallsicherheit	692
Sicherheit der Infrastruktur	693
Verwendung von -Schnittstellen-VPC-Endpunkten	694
Konfigurations- und Schwachstellenanalyse für Amazon Keyspaces	701
Bewährte Methoden für die Gewährleistung der Sicherheit	701
Bewährte Methoden für vorbeugende Sicherheitsmaßnahmen	701
Bewährte Methoden für aufdeckende Sicherheitsmaßnahmen	703
CQL-Sprachreferenz	706
Sprachelemente	706

IDs	706
Konstanten	707
Bedingungen	707
Datentypen	707
JSON-Kodierung von Amazon Keyspaces-Datentypen	712
DDL-Anweisungen	715
Keyspaces	716
Tabellen	719
DML-Anweisungen.	732
SELECT	733
INSERT	736
UPDATE	738
DELETE	739
Integrierte Funktionen	740
Skalarfunktionen	740
Kontingente	743
Service Quotas für Amazon Keyspaces	743
Erhöhen oder Verringern des Durchsatzes (für Tabellen im Modus bereitgestellter Kapazität) ..	749
Erhöhen des bereitgestellten Durchsatzes	749
Senken des bereitgestellten Durchsatzes	749
Amazon Keyspaces-Verschlüsselung im Ruhezustand	749
Dokumentverlauf	751
.....	dcclxiii

Was ist Amazon Keyspaces (für Apache Cassandra)?

Amazon Keyspaces (für Apache Cassandra) ist ein skalierbarer, hochverfügbarer und verwalteter Apache Cassandra-kompatibler Datenbankservice. Mit Amazon Keyspaces müssen Sie keine Server bereitstellen, patchen oder verwalten und Sie müssen keine Software installieren, warten oder betreiben.

Amazon Keyspaces ist serverlos, sodass Sie nur für die Ressourcen zahlen, die Sie nutzen, und der Service skaliert Tabellen automatisch als Reaktion auf den Anwendungsverkehr nach oben und unten. Sie können Anwendungen erstellen, die Tausende von Anfragen pro Sekunde mit praktisch unbegrenztem Durchsatz und Speicherplatz bearbeiten.

Note

Apache Cassandra ist ein Open-Source-Datenspeicher mit vielen Spalten, der für die Verarbeitung großer Datenmengen konzipiert ist. [Weitere Informationen finden Sie unter Apache Cassandra.](#)

Amazon Keyspaces macht es einfach, Cassandra-Workloads in der zu migrieren, auszuführen und zu skalieren. AWS Cloud Mit nur wenigen Klicks auf der AWS Management Console oder ein paar Codezeilen können Sie Schlüsselräume und Tabellen in Amazon Keyspaces erstellen, ohne Infrastruktur bereitstellen oder Software installieren zu müssen.

Mit Amazon Keyspaces können Sie Ihre bestehenden Cassandra-Workloads mit demselben Cassandra-Anwendungscode und denselben Entwicklertools ausführen, die Sie heute AWS verwenden.

Eine Liste der verfügbaren Endpunkte AWS-Regionen und Endpunkte finden Sie unter [Service-Endpunkte für Amazon Keyspaces](#).

Wir empfehlen Ihnen, zunächst die folgenden Abschnitte zu lesen:

Themen

- [Amazon Keyspaces: So funktioniert's](#)
- [Anwendungsfälle von Amazon Keyspaces](#)
- [Was ist Cassandra Query Language \(CQL\)?](#)

Amazon Keyspaces: So funktioniert's

Amazon Keyspaces beseitigt den administrativen Aufwand für die Verwaltung von Cassandra. Um zu verstehen, warum, ist es hilfreich, mit der Cassandra-Architektur zu beginnen und sie dann mit Amazon Keyspaces zu vergleichen.

Themen

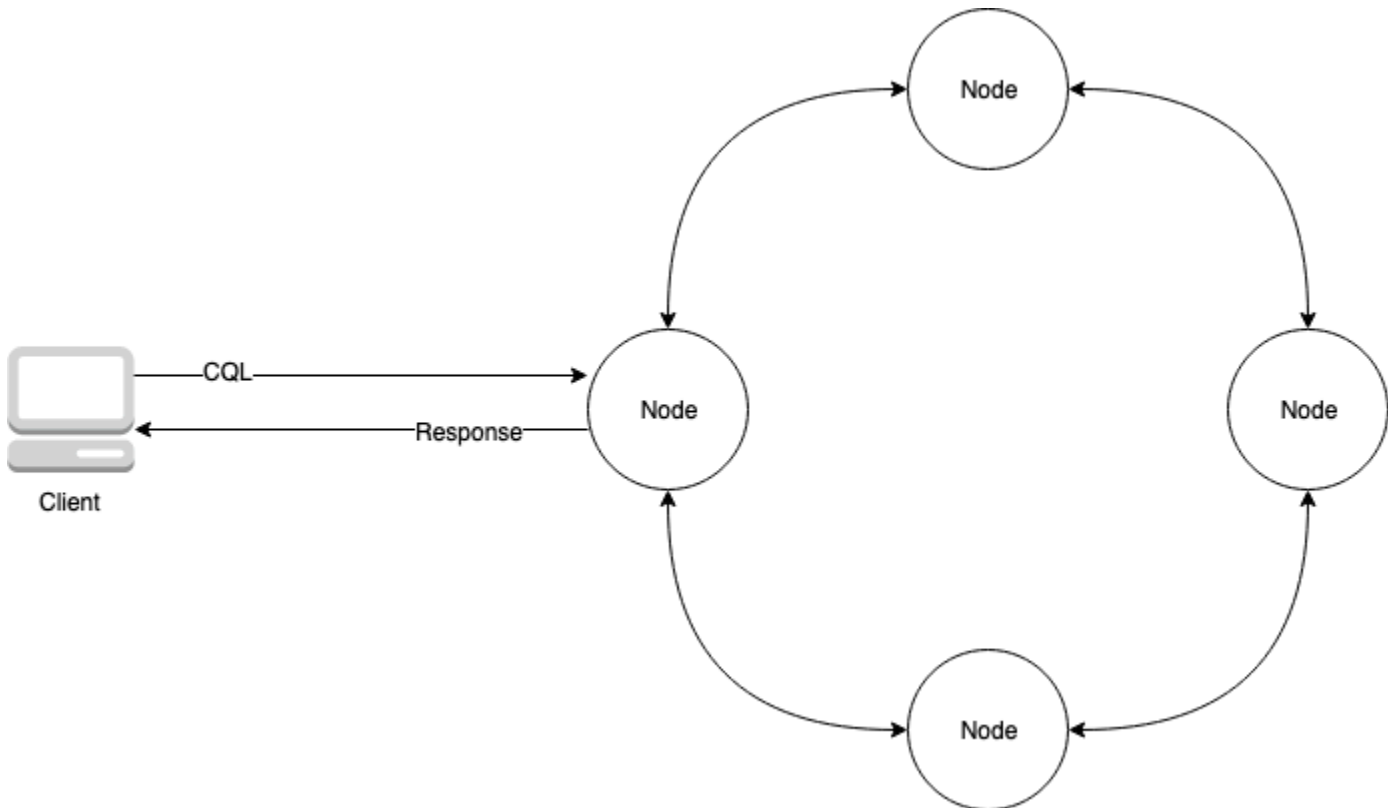
- [Architektur auf hohem Niveau: Apache Cassandra im Vergleich zu Amazon Keyspaces](#)
- [Cassandra-Datenmodell](#)
- [Zugreifen auf Amazon Keyspaces von einer Anwendung aus](#)

Architektur auf hohem Niveau: Apache Cassandra im Vergleich zu Amazon Keyspaces

Herkömmliches Apache Cassandra wird in einem Cluster bereitgestellt, der aus einem oder mehreren Knoten besteht. Sie sind dafür verantwortlich, jeden Knoten zu verwalten und Knoten hinzuzufügen und zu entfernen, wenn Ihr Cluster skaliert.

Ein Client-Programm greift auf Cassandra zu, indem es eine Verbindung zu einem der Knoten herstellt und Cassandra Query Language (CQL) -Anweisungen ausgibt. CQL ähnelt SQL, der beliebten Sprache, die in relationalen Datenbanken verwendet wird. Obwohl Cassandra keine relationale Datenbank ist, bietet CQL eine vertraute Oberfläche zum Abfragen und Bearbeiten von Daten in Cassandra.

Das folgende Diagramm zeigt einen einfachen Apache Cassandra-Cluster, der aus vier Knoten besteht.



Eine Cassandra-Produktionsumgebung kann aus Hunderten von Knoten bestehen, die auf Hunderten von physischen Computern in einem oder mehreren physischen Rechenzentren ausgeführt werden. Dies kann zu einer betrieblichen Belastung für Anwendungsentwickler führen, die neben der Installation, Wartung und dem Betrieb von Software auch Server bereitstellen, patchen und verwalten müssen.

Mit Amazon Keyspaces (für Apache Cassandra) müssen Sie keine Server bereitstellen, patchen oder verwalten, sodass Sie sich auf die Entwicklung besserer Anwendungen konzentrieren können. Amazon Keyspaces bietet zwei Durchsatzkapazitätsmodi für Lese- und Schreibvorgänge: auf Abruf und bereitgestellt. Sie können den Durchsatzkapazitätsmodus Ihrer Tabelle wählen, um den Preis für Lese- und Schreibvorgänge auf der Grundlage der Vorhersagbarkeit und Variabilität Ihrer Arbeitslast zu optimieren.

Im On-Demand-Modus zahlen Sie nur für die Lese- und Schreibvorgänge, die Ihre Anwendung tatsächlich ausführt. Sie müssen die Durchsatzkapazität Ihrer Tabelle nicht im Voraus angeben. Amazon Keyspaces passt Ihren Anwendungsdatenverkehr fast sofort an, wenn er steigt oder fällt, und ist somit eine gute Option für Anwendungen mit unvorhersehbarem Datenverkehr.

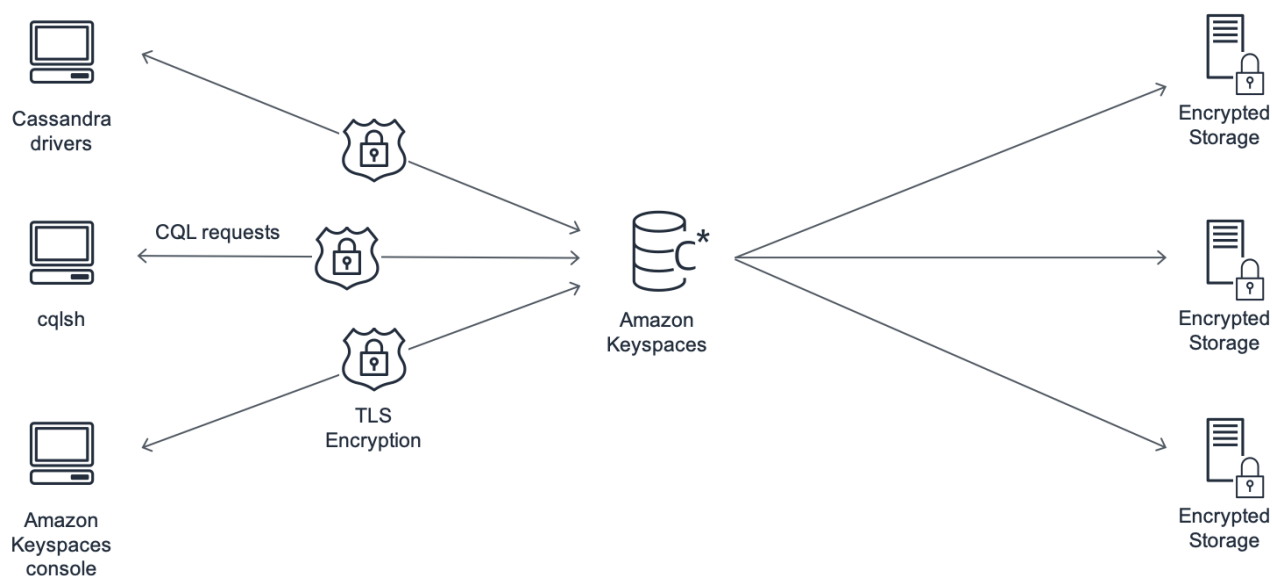
Der Modus „Bereitgestellte Kapazität“ hilft Ihnen, den Preis für den Durchsatz zu optimieren, wenn Sie einen vorhersehbaren Anwendungsdatenverkehr haben und den Kapazitätsbedarf Ihrer Tabelle

im Voraus prognostizieren können. Im Modus „Bereitgestellte Kapazität“ geben Sie die Anzahl der Lese- und Schreibvorgänge pro Sekunde an, die Ihre Anwendung voraussichtlich ausführen soll. Sie können die bereitgestellte Kapazität für Ihre Tabelle automatisch erhöhen und verringern, indem Sie die [automatische](#) Skalierung aktivieren.

Sie können den Kapazitätsmodus Ihrer Tabelle einmal täglich ändern, wenn Sie mehr über die Datenverkehrsmuster Ihres Workloads erfahren oder wenn Sie mit einem starken Anstieg des Datenverkehrs rechnen, z. B. aufgrund eines wichtigen Ereignisses, von dem Sie erwarten, dass es zu einem starken Tabellenverkehr kommen wird. Weitere Informationen zur Bereitstellung von Lese- und Schreibkapazität finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#)

Amazon Keyspaces (für Apache Cassandra) speichert drei Kopien Ihrer Daten in mehreren [Availability Zones, um Haltbarkeit und hohe Verfügbarkeit](#) zu gewährleisten. Darüber hinaus profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die auf die Anforderungen der sicherheitssensibelsten Unternehmen zugeschnitten ist. Die Verschlüsselung im Ruhezustand wird automatisch aktiviert, wenn Sie eine neue Amazon Keyspaces-Tabelle erstellen und für alle Client-Verbindungen Transport Layer Security (TLS) erforderlich ist. Zu den zusätzlichen AWS Sicherheitsfunktionen gehören [Überwachung](#) und [Virtual Private Cloud \(VPC\) -Endpunkte](#). [AWS Identity and Access Management](#) Eine Übersicht über alle verfügbaren Sicherheitsfunktionen finden Sie unter [Sicherheit](#)

Das folgende Diagramm zeigt die Architektur von Amazon Keyspaces.



Ein Client-Programm greift auf Amazon Keyspaces zu, indem es sich mit einem vordefinierten Endpunkt (Hostname und Portnummer) verbindet und CQL-Anweisungen ausgibt. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called “Service-Endpunkte”](#)

Cassandra-Datenmodell

Wie Sie Ihre Daten für Ihren Geschäftsszenario modellieren, ist entscheidend, um eine optimale Leistung von Amazon Keyspaces zu erzielen. Ein schlechtes Datenmodell kann die Leistung erheblich beeinträchtigen.

Obwohl CQL ähnlich aussieht wie SQL, sind die Backends von Cassandra und relationalen Datenbanken sehr unterschiedlich und müssen unterschiedlich angegangen werden. Im Folgenden sind einige der wichtigsten Aspekte aufgeführt, die es zu berücksichtigen gilt:

Speicherung

Sie können Ihre Cassandra-Daten in Tabellen visualisieren, wobei jede Zeile einen Datensatz und jede Spalte ein Feld innerhalb dieses Datensatzes darstellt.

Tabellendesign: Zuerst abfragen

In CQL gibt es keine JOIN s. Daher sollten Sie beim Entwerfen Ihrer Tabellen die Form Ihrer Daten und die Art und Weise, wie Sie für Ihre geschäftlichen Anwendungsfälle darauf zugreifen müssen, berücksichtigen. Dies kann zu einer De-normalisierung mit duplizierten Daten führen. Sie sollten jede Ihrer Tabellen speziell für ein bestimmtes Zugriffsmuster entwerfen.

Partitionen

Ihre Daten werden in Partitionen auf der Festplatte gespeichert. Die Anzahl der Partitionen, in denen Ihre Daten gespeichert sind, und wie sie auf die Partitionen verteilt sind, wird durch Ihren Partitionsschlüssel bestimmt. Die Art und Weise, wie Sie Ihren Partitionsschlüssel definieren, kann erhebliche Auswirkungen auf die Leistung Ihrer Abfragen haben. Bewährte Methoden finden Sie unter [the section called “Entwerfen von Partitionsschlüsseln”](#).

Primärschlüssel

In Cassandra werden Daten als Schlüssel-Wert-Paar gespeichert. Zu diesem Zweck muss jede Cassandra-Tabelle einen Primärschlüssel haben, der der Schlüssel zu jeder Zeile in der Tabelle ist. Der Primärschlüssel besteht aus einem erforderlichen Partitionsschlüssel und optionalen Clusterspalten. Die Daten, aus denen der Primärschlüssel besteht, müssen für alle Datensätze in einer Tabelle eindeutig sein.

- **Partitionsschlüssel** — Der Partitionsschlüsselteil des Primärschlüssels ist erforderlich und bestimmt, in welcher Partition Ihres Clusters die Daten gespeichert werden. Der Partitionsschlüssel kann eine einzelne Spalte oder ein zusammengesetzter Wert sein, der aus zwei oder mehr Spalten besteht. Sie würden einen zusammengesetzten Partitionsschlüssel verwenden, wenn ein einspaltiger Partitionsschlüssel dazu führen würde, dass eine einzelne Partition oder einige wenige Partitionen die meisten Daten enthalten und somit die meisten Festplatten-I/O-Operationen ausführen.
- **Clusterspalte** — Die optionale Gruppierungsspalte Ihres Primärschlüssels bestimmt, wie die Daten innerhalb der einzelnen Partitionen gruppiert und sortiert werden. Wenn Sie eine Clusterspalte in Ihren Primärschlüssel aufnehmen, kann die Clustering-Spalte eine oder mehrere Spalten enthalten. Wenn die Gruppierungsspalte mehrere Spalten enthält, wird die Sortierreihenfolge von links nach rechts durch die Reihenfolge bestimmt, in der die Spalten in der Gruppierungsspalte aufgeführt sind.

Weitere Informationen zum NoSQL-Design und zu Amazon Keyspaces finden Sie unter [the section called “NoSQL-Design”](#). Weitere Informationen zu Amazon Keyspaces und Datenmodellierung finden Sie unter [the section called “Datenmodellierung”](#).

Zugreifen auf Amazon Keyspaces von einer Anwendung aus

Amazon Keyspaces (für Apache Cassandra) implementiert die Apache Cassandra Query Language (CQL) -API, sodass Sie CQL- und Cassandra-Treiber verwenden können, die Sie bereits verwenden. Das Aktualisieren Ihrer Anwendung ist so einfach wie das Aktualisieren Ihres Cassandra-Treibers oder Ihrer `cqlsh` Konfiguration, sodass sie auf den Amazon Keyspaces-Serviceendpunkt verweisen. Weitere Informationen zu den erforderlichen Anmeldeinformationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#)

Note

Um Ihnen den Einstieg zu erleichtern, finden Sie end-to-end Codebeispiele für die Verbindung zu Amazon Keyspaces mithilfe verschiedener Cassandra-Client-Treiber im Amazon Keyspaces-Codebeispiel-Repository unter [GitHub](#)

Stellen Sie sich das folgende Python-Programm vor, das eine Verbindung zu einem Cassandra-Cluster herstellt und eine Tabelle abfragt.

```
from cassandra.cluster import Cluster
#TLS/SSL configuration goes here

ksp = 'MyKeyspace'
tbl = 'WeatherData'

cluster = Cluster(['NNN.NNN.NNN.NNN'], port=NNNN)
session = cluster.connect(ksp)

session.execute('USE ' + ksp)

rows = session.execute('SELECT * FROM ' + tbl)
for row in rows:
    print(row)
```

Um dasselbe Programm für Amazon Keyspaces auszuführen, müssen Sie:

- Fügen Sie den Cluster-Endpunkt und den Port hinzu: Beispielsweise kann der Host durch einen Service-Endpunkt ersetzt werden, z. B. `cassandra.us-east-2.amazonaws.com` und die Portnummer durch:9142.
- Fügen Sie die TLS/SSL-Konfiguration hinzu: Weitere Informationen zum Hinzufügen der TLS/SSL-Konfiguration für die Verbindung zu Amazon Keyspaces mithilfe eines Cassandra-Client-Python-Treibers finden Sie unter [Verwenden eines Cassandra-Python-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)

Anwendungsfälle von Amazon Keyspaces

Im Folgenden sind nur einige der Möglichkeiten aufgeführt, wie Sie Amazon Keyspaces verwenden können:

- Entwickeln Sie Anwendungen, die eine geringe Latenz erfordern — Verarbeiten Sie Daten mit hoher Geschwindigkeit für Anwendungen, die single-digit-millisecond Latenz erfordern, z. B. für die Wartung von Industrieanlagen, die Handelsüberwachung, das Flottenmanagement und die Routenoptimierung.
- Erstellen Sie Anwendungen mit Open-Source-Technologien — Erstellen Sie Anwendungen AWS mithilfe von Open-Source-Cassandra-APIs und -Treibern, die für eine Vielzahl von Programmiersprachen wie Java, Python, Ruby, Microsoft.NET, Node.js, PHP, C++, Perl und Go verfügbar sind. Codebeispiele finden Sie unter [Bibliotheken und Tools](#).

- Verschieben Sie Ihre Cassandra-Workloads in die Cloud — Cassandra-Tabellen selbst zu verwalten ist zeitaufwändig und teuer. Mit Amazon Keyspaces können Sie Cassandra-Tabellen einrichten, sichern und skalieren, AWS Cloud ohne die Infrastruktur verwalten zu müssen. Weitere Informationen finden Sie unter [Serverloses Ressourcenmanagement](#).

Was ist Cassandra Query Language (CQL)?

Cassandra Query Language (CQL) ist die Hauptsprache für die Kommunikation mit Apache Cassandra. Amazon Keyspaces (für Apache Cassandra) ist mit der CQL 3.x-API kompatibel (abwärtskompatibel mit Version 2.x).

Um CQL-Abfragen auszuführen, können Sie einen der folgenden Schritte ausführen:

- Verwenden Sie den CQL-Editor auf dem AWS Management Console
- Verwenden Sie AWS CloudShell und die [Cqlsh-Erweiterung](#).
- Führen Sie sie auf dem Client aus. `cqlsh`
- Führen Sie sie programmgesteuert mit einem für Apache 2.0 lizenzierten Cassandra-Clienttreiber aus.

Darüber hinaus können Sie mit dem AWS SDK und dem auf Amazon Keyspaces zugreifen. AWS Command Line Interface

Weitere Informationen zur Verwendung dieser Methoden für den Zugriff auf Amazon Keyspaces finden Sie unter [Zugreifen auf Amazon Keyspaces \(für Apache Cassandra\)](#).

Weitere Informationen zu CQL finden Sie unter [CQL-Sprachreferenz für Amazon Keyspaces \(für Apache Cassandra\)](#)

Wie ist Amazon Keyspaces (für Apache Cassandra) im Vergleich zu Apache Cassandra?

Um eine Verbindung zu Amazon Keyspaces herzustellen, können Sie entweder einen öffentlichen oder einen [AWS privaten Endpunkt](#) mithilfe von [Interface VPC-Endpunkten \(AWS PrivateLink\)](#) in der [Amazon Virtual Private Cloud](#) verwenden. Je nach verwendetem Endpunkt kann Amazon Keyspaces dem Client auf eine der folgenden Arten angezeigt werden.

AWS Verbindung zum Service-Endpunkt

Dies ist eine Verbindung, die über einen beliebigen [öffentlichen Endpunkt](#) hergestellt wird. In diesem Fall erscheint Amazon Keyspaces dem Client als Apache Cassandra 3.11.2-Cluster mit neun Knoten.

Schnittstelle VPC-Endpunktverbindung

Dies ist eine private Verbindung, die über einen [VPC-Schnittstellen-Endpunkt](#) hergestellt wird. In diesem Fall erscheint Amazon Keyspaces dem Client als Apache Cassandra 3.11.2-Cluster mit drei Knoten.

Unabhängig vom Verbindungstyp und der Anzahl der Knoten, die für den Client sichtbar sind, bietet Amazon Keyspaces praktisch unbegrenzten Durchsatz und Speicherplatz. Zu diesem Zweck ordnet Amazon Keyspaces die Knoten Load Balancern zu, die Ihre Abfragen an eine der vielen zugrunde liegenden Speicherpartitionen weiterleiten. Weitere Informationen zu Verbindungen finden Sie unter [the section called “Wie funktionieren sie”](#).

Amazon Keyspaces speichert Daten in Partitionen. Eine Partition ist eine Speicherzuweisung für eine Tabelle, die von Solid-State-Laufwerken (SSDs) unterstützt wird. Amazon Keyspaces repliziert Ihre Daten automatisch innerhalb mehrerer [Availability Zones](#), AWS-Region um Haltbarkeit und hohe Verfügbarkeit zu gewährleisten. Wenn Ihr Durchsatz- oder Speicherbedarf steigt, übernimmt Amazon Keyspaces die Partitionsverwaltung für Sie und stellt automatisch die erforderlichen zusätzlichen Partitionen bereit. Weitere Informationen finden Sie unter [the section called “Speicher”](#).

Amazon Keyspaces unterstützt alle häufig verwendeten Cassandra-Datenebenenoperationen, z. B. das Erstellen von Schlüsselräumen und Tabellen, das Lesen von Daten und das Schreiben von Daten. Amazon Keyspaces ist [serverlos](#), sodass Sie keine Server bereitstellen, patchen oder verwalten müssen. Sie müssen auch keine Software installieren, warten oder betreiben. Daher

müssen Sie in Amazon Keyspaces nicht die API-Operationen der Cassandra-Steuerungsebene verwenden, um Cluster- und Knoteneinstellungen zu verwalten.

Amazon Keyspaces konfiguriert automatisch Einstellungen wie den Replikationsfaktor und das Konsistenzniveau, um Ihnen hohe Verfügbarkeit, Haltbarkeit und single-digit-millisecond Leistung zu bieten. [Für noch mehr Stabilität und lokale Lesevorgänge mit geringer Latenz bietet Amazon Keyspaces Replikation in mehreren Regionen.](#)

Themen

- [Funktionale Unterschiede: Amazon Keyspaces im Vergleich zu Apache Cassandra](#)
- [Unterstützte Cassandra-APIs, Operationen, Funktionen und Datentypen in Amazon Keyspaces](#)
- [Unterstützte Apache Cassandra-Konsistenzstufen in Amazon Keyspaces](#)

Funktionale Unterschiede: Amazon Keyspaces im Vergleich zu Apache Cassandra

Im Folgenden sind die funktionalen Unterschiede zwischen Amazon Keyspaces und Apache Cassandra aufgeführt.

Themen

- [APIs, Operationen und Datentypen von Apache Cassandra](#)
- [Asynchrones Erstellen und Löschen von Schlüsselräumen und Tabellen](#)
- [Authentifizierung und Autorisierung](#)
- [Stapel](#)
- [Cluster-Konfiguration](#)
- [Verbindungen](#)
- [INSchlüsselwort](#)
- [Optimierung des Durchsatzes für CQL-Abfragen](#)
- [FROZENSammlungen](#)
- [Einfache Transaktionen](#)
- [Load Balancing](#)
- [Paginierung](#)
- [Partitionierer](#)

- [Vorbereitete Aussagen](#)
- [Bereich löschen](#)
- [Systemtabellen](#)
- [Zeitstempel](#)

APIs, Operationen und Datentypen von Apache Cassandra

Amazon Keyspaces unterstützt alle häufig verwendeten Cassandra-Datenebenenoperationen, z. B. das Erstellen von Schlüsselräumen und Tabellen, das Lesen von Daten und das Schreiben von Daten. Informationen zu den zurzeit unterstützten Komponenten finden Sie unter [Unterstützte Cassandra-APIs, Operationen, Funktionen und Datentypen in Amazon Keyspaces](#).

Asynchrones Erstellen und Löschen von Schlüsselräumen und Tabellen

Amazon Keyspaces führt DDL-Operationen (Data Definition Language), wie z. B. das Erstellen und Löschen von Schlüsselräumen und Tabellen, asynchron aus. Informationen zur Überwachung des Erstellungstatus von Ressourcen finden Sie unter und. [the section called “Erstellen von Keyspaces”](#) [the section called “Erstellen von Tabellen”](#) Eine Liste der DDL-Anweisungen in der CQL-Sprachreferenz finden Sie unter. [the section called “DDL-Anweisungen”](#)

Authentifizierung und Autorisierung

Amazon Keyspaces (für Apache Cassandra) verwendet AWS Identity and Access Management (IAM) für die Benutzerauthentifizierung und -autorisierung und unterstützt die entsprechenden Autorisierungsrichtlinien wie Apache Cassandra. Daher unterstützt Amazon Keyspaces die Sicherheitskonfigurationsbefehle von Apache Cassandra nicht.

Stapel

Amazon Keyspaces unterstützt nicht protokollierte Batch-Befehle mit bis zu 30 Befehlen pro Batch. In einem Batch sind nur bedingungslose INSERT DELETE Oder-Befehle zulässig. UPDATE Protokolierte Batches werden nicht unterstützt.

Cluster-Konfiguration

Amazon Keyspaces ist serverlos, sodass keine Cluster, Hosts oder Java Virtual Machines (JVMs) konfiguriert werden müssen. Cassandras Einstellungen für Komprimierung, Komprimierung, Caching,

Garbage-Collection und Bloom-Filterung gelten nicht für Amazon Keyspaces und werden ignoriert, falls angegeben.

Verbindungen

Sie können vorhandene Cassandra-Treiber für die Kommunikation mit Amazon Keyspaces verwenden, müssen die Treiber jedoch anders konfigurieren. Amazon Keyspaces unterstützt bis zu 3.000 CQL-Abfragen pro TCP-Verbindung pro Sekunde, aber die Anzahl der Verbindungen, die ein Treiber herstellen kann, ist unbegrenzt.

Die meisten Open-Source-Cassandra-Treiber richten einen Verbindungspool zu Cassandra ein und verteilen Abfragen über diesen Verbindungspool. Amazon Keyspaces stellt Treibern 9 Peer-IP-Adressen zur Verfügung, und das Standardverhalten der meisten Treiber besteht darin, eine einzige Verbindung zu jeder Peer-IP-Adresse herzustellen. Daher beträgt der maximale CQL-Abfragedurchsatz eines Treibers, der die Standardeinstellungen verwendet, 27.000 CQL-Abfragen pro Sekunde.

Um diese Zahl zu erhöhen, empfehlen wir, die Anzahl der Verbindungen pro IP-Adresse zu erhöhen, die Ihr Treiber in seinem Verbindungspool verwaltet. Wenn Sie beispielsweise die maximale Anzahl an Verbindungen pro IP-Adresse auf 2 festlegen, verdoppelt sich der maximale Durchsatz Ihres Treibers auf 54.000 CQL-Abfragen pro Sekunde.

Als bewährte Methode empfehlen wir, Treiber so zu konfigurieren, dass sie 500 CQL-Abfragen pro Sekunde und Verbindung verwenden, um Mehraufwand zu vermeiden und die Verteilung zu verbessern. In diesem Szenario sind für die Planung von 18.000 CQL-Abfragen pro Sekunde 36 Verbindungen erforderlich. Die Konfiguration des Treibers für 4 Verbindungen an 9 Endpunkten ermöglicht 36 Verbindungen, die 500 Anfragen pro Sekunde ausführen. Weitere Informationen zu bewährten Methoden für Verbindungen finden Sie unter [the section called “Verbindungen”](#).

Bei der Verbindung mit VPC-Endpunkten sind möglicherweise weniger Endpunkte verfügbar. Das bedeutet, dass Sie die Anzahl der Verbindungen in der Treiberkonfiguration erhöhen müssen. Weitere Informationen zu bewährten Methoden für VPC-Verbindungen finden Sie unter [the section called “VPC-Endpunktverbindungen”](#).

IN Schlüsselwort

Amazon Keyspaces unterstützt das IN Schlüsselwort in der SELECT Anweisung. IN wird mit UPDATE und DELETE nicht unterstützt. Wenn das IN Schlüsselwort in der SELECT Anweisung verwendet wird,

werden die Ergebnisse der Abfrage in der Reihenfolge zurückgegeben, in der die Schlüssel in der SELECT Anweisung dargestellt werden. In Cassandra sind die Ergebnisse lexikografisch geordnet.

Bei Verwendung `ORDER BY` wird eine vollständige Neuordnung mit deaktivierter Paginierung nicht unterstützt und die Ergebnisse werden innerhalb einer Seite angeordnet. Slice-Abfragen werden mit dem Schlüsselwort nicht unterstützt. `IN TOKENS` werden mit dem `IN` Schlüsselwort nicht unterstützt. Amazon Keyspaces verarbeitet Abfragen mit dem `IN` Schlüsselwort, indem es Unterabfragen erstellt. Jede Unterabfrage zählt als Verbindung im Rahmen der Obergrenze von 3.000 CQL-Abfragen pro TCP-Verbindung pro Sekunde. Weitere Informationen finden Sie unter [the section called “INSELECTAussage”](#).

Optimierung des Durchsatzes für CQL-Abfragen

Amazon Keyspaces unterstützt bis zu 3.000 CQL-Abfragen pro TCP-Verbindung pro Sekunde, aber die Anzahl der Verbindungen, die ein Treiber herstellen kann, ist unbegrenzt.

Die meisten Open-Source-Cassandra-Treiber richten einen Verbindungspool zu Cassandra ein und verteilen Abfragen über diesen Verbindungspool. Amazon Keyspaces stellt Treibern 9 Peer-IP-Adressen zur Verfügung, und das Standardverhalten der meisten Treiber besteht darin, eine einzige Verbindung zu jeder Peer-IP-Adresse herzustellen. Daher beträgt der maximale CQL-Abfragedurchsatz eines Treibers, der die Standardeinstellungen verwendet, 27.000 CQL-Abfragen pro Sekunde.

Um diese Zahl zu erhöhen, empfehlen wir, die Anzahl der Verbindungen pro IP-Adresse zu erhöhen, die Ihr Treiber in seinem Verbindungspool verwaltet. Wenn Sie beispielsweise die maximale Anzahl an Verbindungen pro IP-Adresse auf 2 festlegen, wird der maximale Durchsatz Ihres Treibers auf 54.000 CQL-Abfragen pro Sekunde verdoppelt.

Weitere Informationen zu bewährten Methoden für Verbindungen finden Sie unter [the section called “Verbindungen”](#)

Bei der Verbindung mit VPC-Endpunkten sind weniger Endpunkte verfügbar. Das bedeutet, dass Sie die Anzahl der Verbindungen in der Treiberkonfiguration erhöhen müssen. Weitere Informationen zu bewährten Methoden für VPC-Endpunktverbindungen finden Sie unter [the section called “VPC-Endpunktverbindungen”](#).

FROZENSammlungen

Das FROZEN Schlüsselwort in Cassandra serialisiert mehrere Komponenten eines Sammlungsdatentyps zu einem einzigen unveränderlichen Wert, der wie ein BLOB behandelt wird. BLOB INSERT und UPDATE Anweisungen überschreiben die gesamte Sammlung.

Amazon Keyspaces unterstützt standardmäßig bis zu fünf Verschachtelungsebenen für eingefrorene Sammlungen. Weitere Informationen finden Sie unter [the section called “Service Quotas für Amazon Keyspaces”](#).

Amazon Keyspaces unterstützt keine Ungleichheitsvergleiche, bei denen die gesamte eingefrorene Sammlung in einer Bedingung UPDATE oder SELECT Anweisung verwendet wird. Das Verhalten für Sammlungen und eingefrorene Sammlungen ist in Amazon Keyspaces dasselbe.

Wenn Sie eingefrorene Sammlungen mit clientseitigen Zeitstempeln verwenden und der Zeitstempel eines Schreibvorgangs dem Zeitstempel einer vorhandenen Spalte entspricht, die nicht abgelaufen oder als veraltet markiert ist, führt Amazon Keyspaces keine Vergleiche durch. Stattdessen kann der Server den neuesten Writer ermitteln, und der letzte Writer gewinnt.

Weitere Informationen zu eingefrorenen Sammlungen finden Sie unter [the section called “Sammlungstypen”](#).

Einfache Transaktionen

Amazon Keyspaces (für Apache Cassandra) unterstützt vollständig die Funktionen zum Vergleichen und Einstellen von DELETE Befehlen, und INSERTUPDATE, die in Apache Cassandra als Lightweight Transactions (LWTs) bezeichnet werden. Als serverloses Angebot bietet Amazon Keyspaces (für Apache Cassandra) gleichbleibende Leistung in jeder Größenordnung, auch für einfache Transaktionen. Mit Amazon Keyspaces gibt es keine Leistungseinbußen bei der Verwendung einfacher Transaktionen.

Load Balancing

Die `system.peers` Tabelleneinträge entsprechen den Load Balancern von Amazon Keyspaces. Um optimale Ergebnisse zu erzielen, empfehlen wir, eine Round-Robin-Load-Balancing-Richtlinie zu verwenden und die Anzahl der Verbindungen pro IP an die Anforderungen Ihrer Anwendung anzupassen.

Paginierung

Amazon Keyspaces paginiert Ergebnisse auf der Grundlage der Anzahl der Zeilen, die es liest, um eine Anfrage zu verarbeiten, und nicht auf der Anzahl der Zeilen, die in der Ergebnismenge zurückgegeben werden. Daher können einige Seiten weniger Zeilen enthalten, als Sie unter `SEITENGRÖSSE` für gefilterte Abfragen angeben. Darüber hinaus paginiert Amazon Keyspaces die Ergebnisse nach dem Lesen von 1 MB Daten automatisch, um Kunden eine konsistente Leseleistung im einstelligen Millisekundenbereich zu bieten. Weitere Informationen finden Sie unter [the section called “Ergebnisse paginieren”](#).

Partitionierer

Die Standard-Partitionierung in Amazon Keyspaces ist Cassandra-Compatible.

`Murmur3Partitioner` Darüber hinaus haben Sie die Wahl, entweder die Amazon Keyspaces `DefaultPartitioner` oder die `RandomPartitioner` Cassandra-kompatiblen zu verwenden.

Mit Amazon Keyspaces können Sie den Partitionierer für Ihr Konto sicher ändern, ohne Ihre Amazon Keyspaces-Daten neu laden zu müssen. Nach Abschluss der Konfigurationsänderung, die etwa 10 Minuten dauert, sehen Kunden die neue Partitionierungseinstellung automatisch, wenn sie das nächste Mal eine Verbindung herstellen. Weitere Informationen finden Sie unter [the section called “Mit Partitionierern arbeiten”](#).

Vorbereitete Aussagen

Amazon Keyspaces unterstützt die Verwendung von vorbereiteten Anweisungen für DML-Operationen (Data Manipulation Language), z. B. das Lesen und Schreiben von Daten. Amazon Keyspaces unterstützt derzeit nicht die Verwendung von vorbereiteten Anweisungen für DDL-Operationen (Data Definition Language), wie das Erstellen von Tabellen und Schlüsselräumen. DDL-Operationen müssen außerhalb von vorbereiteten Anweisungen ausgeführt werden.

Bereich löschen

Amazon Keyspaces unterstützt das Löschen von Zeilen im Bereich. Ein Bereich ist ein zusammenhängender Satz von Zeilen innerhalb einer Partition. Sie geben einen Bereich in einer DELETE-Operation mithilfe einer WHERE-Klausel an. Sie können den Bereich als gesamte Partition angeben.

Darüber hinaus können Sie einen Bereich als Teilmenge zusammenhängender Zeilen innerhalb einer Partition angeben, indem Sie relationale Operatoren verwenden (z. B. '>', '<') oder indem Sie

den Partitionsschlüssel angeben und eine oder mehrere Clusterspalten weglassen. Mit Amazon Keyspaces können Sie bis zu 1.000 Zeilen innerhalb eines Bereichs in einem einzigen Vorgang löschen. Darüber hinaus sind Bereichslöschungen atomar, aber nicht isoliert.

Systemtabellen

Amazon Keyspaces füllt die Systemtabellen, die für die Open-Source-Cassandra-Treiber von Apache 2.0 erforderlich sind. Die Systemtabellen, die für einen Client sichtbar sind, enthalten Informationen, die für den authentifizierten Benutzer eindeutig sind. Die Systemtabellen werden vollständig von Amazon Keyspaces gesteuert und sind schreibgeschützt.

Ein schreibgeschützter Zugriff auf Systemtabellen ist erforderlich, und Sie können ihn mithilfe von IAM-Zugriffsrichtlinien steuern. Weitere Informationen finden Sie unter [the section called “Verwalten des Zugriffs mit Richtlinien”](#). Sie müssen tagbasierte Zugriffskontrollrichtlinien für Systemtabellen unterschiedlich definieren, je nachdem, ob Sie die AWS SDK- oder Cassandra Query Language (CQL) -API-Aufrufe über Cassandra-Treiber und Entwicklertools verwenden. Weitere Informationen zur tagbasierten Zugriffskontrolle für Systemtabellen finden Sie unter [the section called “ Amazon Keyspaces-Ressourcenzugriff basierend auf Tags”](#)

Wenn Sie über Amazon [VPC-Endpunkte auf Amazon](#) Keyspaces zugreifen, werden in der `system.peers` Tabelle Einträge für jeden Amazon VPC-Endpunkt angezeigt, für den Amazon Keyspaces Zugriffsberechtigungen besitzt. Infolgedessen gibt Ihr Cassandra-Treiber möglicherweise eine [Warnmeldung](#) über den Kontrollknoten selbst in der Tabelle aus. `system.peers` Sie können diese Warnung getrost ignorieren.

Zeitstempel

In Amazon Keyspaces sind Zeitstempel auf Zellenebene, die mit den Standardzeitstempeln in Apache Cassandra kompatibel sind, eine optionale Funktion.

Die `USING TIMESTAMP` Klausel und die `WRITETIME` Funktion sind nur verfügbar, wenn clientseitige Zeitstempel für eine Tabelle aktiviert sind. Weitere Informationen zu clientseitigen Zeitstempeln in Amazon Keyspaces finden Sie unter [ClientZeitstempel](#)

Unterstützte Cassandra-APIs, Operationen, Funktionen und Datentypen in Amazon Keyspaces

Amazon Keyspaces (für Apache Cassandra) ist mit der Cassandra Query Language (CQL) 3.11-API (abwärtskompatibel mit Version 2.x) kompatibel.

Amazon Keyspaces unterstützt alle häufig verwendeten Cassandra-Datenebenenoperationen, z. B. das Erstellen von Schlüsselräumen und Tabellen, das Lesen von Daten und das Schreiben von Daten.

In den folgenden Abschnitten sind die unterstützten Funktionen aufgeführt.

Themen

- [Cassandra-API-Unterstützung](#)
- [API-Unterstützung für die Cassandra-Steuerebene](#)
- [API-Unterstützung für Cassandra-Datenebene](#)
- [Unterstützung der Cassandra-Funktion](#)
- [Unterstützung des Cassandra-Datentyps](#)

Cassandra-API-Unterstützung

API-Operation	Unterstützt
CREATE KEYSPACE	Ja
ALTER KEYSPACE	Ja
DROP KEYSPACE	Ja
CREATE TABLE	Ja
ALTER TABLE	Ja
DROP TABLE	Ja
CREATE INDEX	Nein
DROP INDEX	Nein
UNLOGGED BATCH	Ja
LOGGED BATCH	Nein
SELECT	Ja

API-Operation	Unterstützt
INSERT	Ja
DELETE	Ja
UPDATE	Ja
USE	Ja
CREATE TYPE	Nein
ALTER TYPE	Nein
DROP TYPE	Nein
CREATE TRIGGER	Nein
DROP TRIGGER	Nein
CREATE FUNCTION	Nein
DROP FUNCTION	Nein
CREATE AGGREGATE	Nein
DROP AGGREGATE	Nein
CREATE MATERIALIZED VIEW	Nein
ALTER MATERIALIZED VIEW	Nein
DROP MATERIALIZED VIEW	Nein
TRUNCATE	Nein

API-Unterstützung für die Cassandra-Steuerebene

Da Amazon Keyspaces verwaltet wird, sind die API-Operationen der Cassandra-Steuerebene zur Verwaltung der Cluster- und Knoteneinstellungen nicht erforderlich. Daher sind die folgenden Cassandra-Funktionen nicht anwendbar.

Funktion	Grund
Umschalten auf Dauerhafte Schreibvorgänge	Alle Schreibvorgänge sind dauerhaft
Lesen Sie die Reperaturereinstellungen	Nicht zutreffend
GC Grace Seconds	Nicht zutreffend
Bloom-Filter-Einstellungen	Nicht zutreffend
Einstellungen für die Komprimierung	Nicht zutreffend
Compression settings (Kompressionseinstellungen)	Nicht zutreffend
Einstellungen für das Zwischenspeichern	Nicht zutreffend
Sicherheitseinstellungen	Ersetzt durch IAM

API-Unterstützung für Cassandra-Datenebene

Funktionsmerkmal	Unterstützt
JSON-Unterstützung für SELECT- und INSERT-Anweisungen	Ja
Statische Spalten	Ja
Zeit bis zum Leben (TTL)	Ja

Unterstützung der Cassandra-Funktion

Weitere Hinweise zu den unterstützten Funktionen finden Sie unter [the section called “Integrierte Funktionen”](#).

Funktion	Unterstützt
Aggregate -Funktionen	Nein
BlobKonvertierung	Ja
Cast	Ja
Datetime-Funktionen	Ja
Funktionen zur Zeitumrechnung	Ja
TimeUuid-Funktionen	Ja
Token	Ja
User defined functions (UDF)	Nein
Uuid	Ja

Unterstützung des Cassandra-Datentyps

Datentyp	Unterstützt	Hinweis
ascii	Ja	
bigint	Ja	
blob	Ja	
boolean	Ja	
counter	Ja	
date	Ja	
decimal	Ja	
double	Ja	

Datentyp	Unterstützt	Hinweis
float	Ja	
frozen	Ja	
inet	Ja	
int	Ja	
list	Ja	
map	Ja	
set	Ja	
smallint	Ja	
text	Ja	
time	Ja	
timestamp	Ja	
timeuuid	Ja	
tinyint	Ja	
tuple	Ja	
user-defined types (UDT)	Nein	Informationen zur Umgestaltung von UDTs mit Protocol Buffers finden Sie unter Amazon Keyspaces Protocol Buffers .
uuid	Ja	
varchar	Ja	
varint	Ja	

Unterstützte Apache Cassandra-Konsistenzstufen in Amazon Keyspaces

Die Themen in diesem Abschnitt beschreiben, welche Apache Cassandra-Konsistenzstufen für Lese- und Schreibvorgänge in Amazon Keyspaces (für Apache Cassandra) unterstützt werden.

Themen

- [Konsistenzstufen schreiben](#)
- [Konsistenzstufen lesen](#)
- [Nicht unterstützte Konsistenzstufen](#)

Konsistenzstufen schreiben

Amazon Keyspaces repliziert alle Schreibvorgänge dreimal in mehreren Availability Zones, um Beständigkeit und hohe Verfügbarkeit zu gewährleisten. Schreibvorgänge werden dauerhaft gespeichert, bevor sie mit dem `LOCAL_QUORUM`-Konsistenzniveau. Für jeden 1-KB-Schreibvorgang wird Ihnen 1 Schreibkapazitätseinheit (WCU) für Tabellen im Bereitstellungsmodus oder 1 Schreibenanforderungseinheit (WRU) für Tabellen im On-Demand-Modus in Rechnung gestellt.

Du kannst `benutzencqlsh` die Konsistenz für alle Abfragen in der aktuellen Sitzung auf festzulegen `LOCAL_QUORUM` mit dem folgenden Code.

```
CONSISTENCY LOCAL_QUORUM;
```

Um die Konsistenzstufe programmgesteuert zu konfigurieren, können Sie die Konsistenz mit den entsprechenden Cassandra-Clienttreibern festlegen. Mit den Java-Treibern der Version 4.x können Sie beispielsweise das Konsistenzniveau in der `app.config`-Datei wie unten gezeigt.

```
basic.request.consistency = LOCAL_QUORUM
```

Wenn Sie einen Java Cassandra-Treiber der Version 3.x verwenden, können Sie die Konsistenzstufe für die Sitzung angeben, indem Sie Folgendes hinzufügen: `withQueryOptions(new QueryOptions().setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM))` wie im folgenden Codebeispiel gezeigt.

```
Session session = Cluster.builder()
```

```

        .addContactPoint(endPoint)
        .withPort(portNumber)
        .withAuthProvider(new SigV4AuthProvider("us-east-2"))
        .withSSL()
        .withQueryOptions(new
QueryOptions().setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM)
        .build())
        .connect();

```

Um die Konsistenzstufe für bestimmte Schreiboperationen zu konfigurieren, können Sie die Konsistenz definieren, wenn Sie `QueryBuilder.insertInt` mit einem `setConsistencyLevelArgument`, wenn Sie den Java-Treiber verwenden.

Konsistenzstufen lesen

Amazon Keyspaces unterstützt drei Lesekonsistenzstufen: `ONE`, `LOCAL_ONE`, und `LOCAL_QUORUM`. Während einer `LOCAL_QUORUM` read, Amazon Keyspaces gibt eine Antwort zurück, die die neuesten Aktualisierungen aller vorherigen erfolgreichen Schreibvorgänge widerspiegelt. Verwendung des Konsistenzniveaus `ONE` oder `LOCAL_ONE` kann die Leistung und Verfügbarkeit Ihrer Leseanfragen verbessern, aber die Antwort spiegelt möglicherweise nicht die Ergebnisse eines kürzlich abgeschlossenen Schreibvorgangs wider.

Für jeden 4 KB gelesenen Wert mit `ONE` oder `LOCAL_ONE` Konsistenz: Ihnen werden 0,5 Lesekapazitätseinheiten (RCUs) für Tabellen im Bereitstellungsmodus oder 0,5 Read Request Units (RRU) für Tabellen im On-Demand-Modus in Rechnung gestellt. Für jeden 4 KB gelesenen Wert mit `LOCAL_QUORUM` Konsistenz: Ihnen wird 1 Lesekapazitätseinheit (RCU) für Tabellen im Bereitstellungsmodus oder 1 Leseanforderungseinheit (RRU) für Tabellen im On-Demand-Modus in Rechnung gestellt.

Die Abrechnung basiert auf der Lesekonsistenz und dem Durchsatzmodus der Lesekapazität pro Tabelle für jeweils 4 KB an Lesevorgängen

Konsistenzgrad	Bereitgestellt	On-Demand-Modus
ONE	0,5 CRU	0,5 RUs
LOCAL_ONE	0,5 CRU	0,5 RUs
LOCAL_QUORUM	1 RCU	1 RRU

Um eine andere Konsistenz für Leseoperationen anzugeben, rufen Sie `QueryBuilder.select` mit einem `setConsistencyLevelArgument`, wenn Sie den Java-Treiber verwenden.

Nicht unterstützte Konsistenzstufen

Die folgenden Konsistenzstufen werden von Amazon Keyspaces nicht unterstützt und führen zu Ausnahmen.

Nicht unterstützte Konsistenzstufen

Apache Cassandra	Amazon Keyspaces
EACH_QUORUM	Nicht unterstützt
QUORUM	Nicht unterstützt
ALL	Nicht unterstützt
TWO	Nicht unterstützt
THREE	Nicht unterstützt
ANY	Nicht unterstützt
SERIAL	Nicht unterstützt
LOCAL_SERIAL	Nicht unterstützt

Zugreifen auf Amazon Keyspaces (für Apache Cassandra)

Sie können über die Konsole auf Amazon Keyspaces zugreifen, programmgesteuert AWS CloudShell, indem Sie einen `cqlsh` Client, das AWS SDK ausführen oder einen für Apache 2.0 lizenzierten Cassandra-Treiber verwenden. Amazon Keyspaces unterstützt Treiber und Clients, die mit Apache Cassandra 3.11.2 kompatibel sind. Bevor Sie auf Amazon Keyspaces zugreifen können, müssen Sie die Einrichtung abschließen AWS Identity and Access Management und Amazon Keyspaces dann Zugriffsberechtigungen für IAM-Identitäten gewähren.

Einrichten AWS Identity and Access Management

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden 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 eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity Center Benutzerhandbuch.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal](#).

Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter [Einen Berechtigungssatz erstellen](#).AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden [Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen](#).

Amazon Keyspaces einrichten

Der Zugriff auf Amazon Keyspaces-Ressourcen wird mithilfe von [IAM](#) verwaltet. Mithilfe von IAM können Sie Richtlinien an IAM-Benutzer, -Rollen und föderierte Identitäten anhängen, die Lese- und Schreibberechtigungen für bestimmte Ressourcen in Amazon Keyspaces gewähren.

Um mit der Erteilung von Berechtigungen für eine IAM-Identität zu beginnen, können Sie eine der AWS verwalteten Richtlinien für Amazon Keyspaces verwenden:

- [AmazonKeyspacesFullAccess](#)— Diese Richtlinie gewährt Berechtigungen für den Zugriff auf alle Ressourcen in Amazon Keyspaces mit vollem Zugriff auf alle Funktionen.
- [AmazonKeyspacesReadOnlyAccess_v2](#) — Diese Richtlinie gewährt Amazon Keyspaces nur Leseberechtigungen.

Eine ausführliche Erläuterung der in den verwalteten Richtlinien definierten Aktionen finden Sie unter [the section called “Von AWS verwaltete Richtlinien”](#)

Um den Umfang der Aktionen, die eine IAM-Identität ausführen kann, oder die Ressourcen einzuschränken, auf die die Identität zugreifen kann, können Sie eine benutzerdefinierte Richtlinie erstellen, die die `AmazonKeyspacesFullAccess` verwaltete Richtlinie als Vorlage verwendet, und alle nicht benötigten Berechtigungen entfernen. Sie können den Zugriff auch auf bestimmte Schlüsselbereiche oder Tabellen einschränken. Weitere Informationen darüber, wie Sie Aktionen einschränken oder den Zugriff auf bestimmte Ressourcen in Amazon Keyspaces einschränken können, finden Sie unter [the section called “So funktioniert Amazon Keyspaces mit IAM”](#).

Um auf Amazon Keyspaces zuzugreifen, nachdem Sie die Richtlinie erstellt AWS-Konto und erstellt haben, die einer IAM-Identität Zugriff auf Amazon Keyspaces gewährt, fahren Sie mit einem der folgenden Abschnitte fort:

- [Verwenden der Konsole](#)
- [Verwenden AWS CloudShell](#)
- [Programmgesteuert verbinden](#)

Zugreifen auf Amazon Keyspaces über die Konsole

Sie können auf die Konsole für Amazon Keyspaces unter <https://console.aws.amazon.com/keyspaces/home> zugreifen. Weitere Informationen zum AWS Management Console Zugriff finden Sie unter [Steuern des Zugriffs von IAM-Benutzern auf das AWS Management Console](#) im IAM-Benutzerhandbuch.

Sie können die Konsole verwenden, um in Amazon Keyspaces Folgendes zu tun:

- Schlüsselräume und Tabellen erstellen, löschen und verwalten.
- Überwachen Sie wichtige Tabellenmetriken auf der Registerkarte Überwachen einer Tabelle:
 - Größe der fakturierbaren Tabelle (Byte)
 - Kapazitätskennzahlen
- Führen Sie Abfragen mit dem CQL-Editor aus, z. B. Daten einfügen, aktualisieren und löschen.
- Ändern Sie die Partitionierungskonfiguration des Kontos.
- Sehen Sie sich die Leistungs- und Fehlermetriken für das Konto im Dashboard an.

Informationen dazu, wie Sie einen Amazon Keyspaces-Schlüsselraum und eine Tabelle erstellen und mit Beispielanwendungsdaten einrichten, finden Sie unter [Erste Schritte mit Amazon Keyspaces \(für Apache Cassandra\)](#)

Für AWS CloudShell den Zugriff auf Amazon Keyspaces verwenden

AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management Console Sie können AWS CLI Befehle für AWS Dienste mit Ihrer bevorzugten Shell (Bash PowerShell oder Z-Shell) ausführen. Um mit Amazon Keyspaces arbeiten zu können `cqlsh`, müssen Sie das `cqlsh-expansion` installieren. `cqlsh-expansion` Installationsanweisungen finden Sie unter [the section called "Verwenden des cqlsh-expansion"](#).

Sie [starten AWS CloudShell von der aus AWS Management Console](#), und die AWS Anmeldeinformationen, mit denen Sie sich an der Konsole angemeldet haben, sind automatisch in einer neuen Shell-Sitzung verfügbar. Diese Vorauthentifizierung von AWS CloudShell Benutzern ermöglicht es Ihnen, die Konfiguration von Anmeldeinformationen zu überspringen, wenn Sie mit AWS Diensten wie Amazon Keyspaces interagieren, die `cqlsh` oder AWS CLI Version 2 (in der Computerumgebung der Shell vorinstalliert) verwenden.

Erhalt von IAM-Berechtigungen für AWS CloudShell

Mithilfe der von bereitgestellten Ressourcen zur Zugriffsverwaltung können Administratoren IAM-Benutzern Berechtigungen erteilen AWS Identity and Access Management, sodass sie auf die Funktionen der Umgebung zugreifen AWS CloudShell und diese nutzen können.

Am schnellsten kann ein Administrator Benutzern Zugriff gewähren, indem er eine AWS verwaltete Richtlinie verwendet. Bei einer [von AWS verwalteten Richtlinie](#) handelt es sich um eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Die folgende AWS verwaltete Richtlinie für CloudShell kann an IAM-Identitäten angehängt werden:

- `AWSCloudShellFullAccess`: Erteilt die Erlaubnis zur Nutzung AWS CloudShell mit vollem Zugriff auf alle Funktionen.

Wenn Sie den Umfang der Aktionen einschränken möchten, die ein IAM-Benutzer ausführen kann AWS CloudShell, können Sie eine benutzerdefinierte Richtlinie erstellen, die die `AWSCloudShellFullAccess` verwaltete Richtlinie als Vorlage verwendet. Weitere Informationen zur Einschränkung der Aktionen, die Benutzern zur Verfügung stehen CloudShell, finden Sie im AWS CloudShell Benutzerhandbuch unter [Verwaltung von AWS CloudShell Zugriff und Nutzung mit IAM-Richtlinien](#).

Note

Für Ihre IAM-Identität ist außerdem eine Richtlinie erforderlich, die Ihnen die Erlaubnis erteilt, Amazon Keyspaces aufzurufen.

Sie können eine AWS verwaltete Richtlinie verwenden, um Ihrer IAM-Identität Zugriff auf Amazon Keyspaces zu gewähren, oder Sie können mit der verwalteten Richtlinie als Vorlage beginnen und die Berechtigungen entfernen, die Sie nicht benötigen. Sie können den Zugriff auch auf bestimmte

Schlüsselbereiche und Tabellen einschränken, um eine benutzerdefinierte Richtlinie zu erstellen. Die folgende verwaltete Richtlinie für Amazon Keyspaces kann an IAM-Identitäten angehängt werden:

- [AmazonKeyspacesFullAccess](#)— Diese Richtlinie gewährt die Erlaubnis zur Nutzung von Amazon Keyspaces mit vollem Zugriff auf alle Funktionen.

Eine ausführliche Erläuterung der in der verwalteten Richtlinie definierten Aktionen finden Sie unter [the section called “Von AWS verwaltete Richtlinien”](#).

Weitere Informationen darüber, wie Sie Aktionen einschränken oder den Zugriff auf bestimmte Ressourcen in Amazon Keyspaces einschränken können, finden Sie unter [the section called “So funktioniert Amazon Keyspaces mit IAM”](#).

Interaktion mit Amazon Keyspaces über AWS CloudShell

Nachdem Sie AWS CloudShell von der aus gestartet haben AWS Management Console, können Sie sofort mit der Interaktion mit Amazon Keyspaces beginnen, indem Sie `cqlsh` oder die Befehlszeilenschnittstelle verwenden. Falls Sie das noch nicht installiert haben `cqlsh-expansion`, finden Sie detaillierte [the section called “Verwenden des cqlsh-expansion”](#) Anweisungen unter.

Note

Wenn Sie den `cqlsh-expansion` in verwenden AWS CloudShell, müssen Sie keine Anmeldeinformationen konfigurieren, bevor Sie Anrufe tätigen, da Sie bereits in der Shell authentifiziert sind.

Connect zu Amazon Keyspaces her und erstellen Sie einen neuen Keyspace. Lesen Sie dann aus einer Systemtabelle, um zu bestätigen, dass der Schlüsselraum erstellt wurde mit AWS CloudShell

1. Von der aus können Sie starten AWS Management Console, CloudShell indem Sie die folgenden Optionen auswählen, die in der Navigationsleiste verfügbar sind:
 - Wählen Sie das CloudShell Symbol.
 - Beginnen Sie mit der Eingabe von „Cloudshell“ in das Suchfeld und wählen Sie dann die CloudShell Option.

2. Mit dem folgenden Befehl können Sie eine Verbindung zu Amazon Keyspaces herstellen. Stellen Sie sicher, dass Sie `cassandra.us-east-1.amazonaws.com` durch den richtigen Endpunkt für Ihre Region ersetzen.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

Wenn die Verbindung erfolgreich ist, sollten Sie eine Ausgabe sehen, die dem folgenden Beispiel ähnelt.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh current consistency level is ONE.
cqlsh>
```

3. Erstellen Sie einen neuen Schlüsselraum mit dem Namen `mykeyspace`. Sie können dazu den folgenden Befehl verwenden.

```
CREATE KEYSPACE mykeyspace WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

4. Um zu bestätigen, dass der Schlüsselraum erstellt wurde, können Sie mit dem folgenden Befehl aus einer Systemtabelle lesen.

```
SELECT * FROM system_schema_mcs.keyspaces WHERE keyspace_name = 'mykeyspace';
```

Wenn der Aufruf erfolgreich ist, zeigt die Befehlszeile eine Antwort des Services an, die der folgenden Ausgabe ähnelt:

```
keyspace_name | durable_writes | replication
-----+-----
+-----+-----
mykeyspace    |                True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}

(1 rows)
```

Programmgesteuertes Herstellen einer Verbindung zu Amazon Keyspaces

In diesem Thema werden die Schritte beschrieben, die erforderlich sind, um programmgesteuert eine Verbindung zu Amazon Keyspaces herzustellen. Es führt Sie durch die Erstellung von IAM-Anmeldeinformationen und listet die verfügbaren Service-Endpunkte auf. AWS Der letzte Abschnitt zeigt, wie Sie mit `cqlsh` eine Verbindung zu Amazon Keyspaces herstellen. [step-by-step Tutorials zum Herstellen einer Verbindung zu Amazon Keyspaces mithilfe verschiedener Apache Cassandra-Treiber](#) finden Sie unter [the section called “Verwenden eines Cassandra-Client-Treibers”](#) Ein step-by-step Tutorial, das zeigt, wie Sie von einem Amazon VPC-Endpunkt aus eine Verbindung zu Amazon Keyspaces herstellen, finden Sie unter [the section called “Verbindung mit VPC-Endpunkten herstellen”](#)

Note

Um Ihnen den Einstieg zu erleichtern, finden Sie end-to-end Codebeispiele für die Verbindung zu Amazon Keyspaces mithilfe verschiedener Cassandra-Client-Treiber im Amazon Keyspaces-Codebeispiel-Repository unter [GitHub](#)

Amazon Keyspaces unterstützt Treiber und Clients, die mit Apache Cassandra 3.11.2 kompatibel sind. Es wird davon ausgegangen, dass Sie die Installationsanweisungen unter bereits abgeschlossen haben. AWS [Zugreifen auf Amazon Keyspaces](#)

Wenn Sie bereits eine haben, lesen Sie die folgenden Themen AWS-Konto, um zu erfahren, wie Sie mithilfe von `cqlsh` programmgesteuert auf Amazon Keyspaces zugreifen können:

Themen

- [Anmeldeinformationen für den programmgesteuerten Zugriff auf Amazon Keyspaces erstellen](#)
- [Service-Endpunkte für Amazon Keyspaces](#)
- [Verwendung `cqlsh`, um eine Verbindung zu Amazon Keyspaces herzustellen](#)
- [Verwendung der AWS CLI](#)
- [Verwenden der API](#)
- [Amazon Keyspaces mit einem AWS SDK verwenden](#)
- [Verwenden eines Cassandra-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)

- [Tutorial: Herstellen einer Verbindung mit Amazon Keyspaces vom Amazon Elastic Kubernetes Service](#)

Anmeldeinformationen für den programmgesteuerten Zugriff auf Amazon Keyspaces erstellen

Um Benutzern und Anwendungen Anmeldeinformationen für den programmatischen Zugriff auf Amazon Keyspaces-Ressourcen zur Verfügung zu stellen, können Sie einen der folgenden Schritte ausführen:

- Erstellen Sie dienstspezifische Anmeldeinformationen, die dem herkömmlichen Benutzernamen und Passwort ähneln, die Cassandra für die Authentifizierung und Zugriffsverwaltung verwendet. AWS dienstspezifische Anmeldeinformationen sind einem bestimmten AWS Identity and Access Management (IAM-) Benutzer zugeordnet und können nur für den Dienst verwendet werden, für den sie erstellt wurden. Weitere Informationen finden Sie unter [Using IAM with Amazon Keyspaces \(für Apache Cassandra\)](#) im IAM-Benutzerhandbuch.

Warning

IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

- Aus Sicherheitsgründen empfehlen wir, IAM-Identitäten zu erstellen, die für alle AWS Dienste verwendet werden, und temporäre Anmeldeinformationen zu verwenden. Das Amazon Keyspaces SigV4-Authentifizierungs-Plugin für Cassandra-Client-Treiber ermöglicht es Ihnen, Anrufe an Amazon Keyspaces mithilfe von IAM-Zugriffsschlüsseln anstelle von Benutzername und Passwort zu authentifizieren. Weitere Informationen darüber, wie das Amazon Keyspaces SigV4-Plugin es [IAM-Benutzern, -Rollen und föderierten Identitäten ermöglicht](#), sich in Amazon Keyspaces-API-Anfragen zu authentifizieren, finden Sie unter [AWS Signature](#) Version 4-Prozess (Sigv4).

Sie können die SigV4-Plugins von den folgenden Orten herunterladen.

- Java: <https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin>.
- Node.js: <https://github.com/aws/aws-sigv4-auth-cassandra-nodejs-driver-plugin>.
- Python: <https://github.com/aws/aws-sigv4-auth-cassandra-python-driver-plugin>.

- Geh: <https://github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin>.

Codebeispiele, die zeigen, wie Verbindungen mithilfe des SigV4-Authentifizierungs-Plug-ins hergestellt werden, finden Sie unter [the section called “Verwenden eines Cassandra-Client-Treibers”](#).

Themen

- [Generieren Sie dienstspezifische Anmeldeinformationen](#)
- [So erstellen und konfigurieren Sie AWS Anmeldeinformationen für Amazon Keyspaces](#)

Generieren Sie dienstspezifische Anmeldeinformationen

Dienstspezifische Anmeldeinformationen ähneln dem herkömmlichen Benutzernamen und Passwort, die Cassandra für die Authentifizierung und Zugriffsverwaltung verwendet. Dienstspezifische Anmeldeinformationen ermöglichen es IAM-Benutzern, auf einen bestimmten Dienst zuzugreifen. AWS Diese langfristigen Anmeldeinformationen können nicht für den Zugriff auf andere AWS Dienste verwendet werden. Sie sind einem bestimmten IAM-Benutzer zugeordnet und können nicht von anderen IAM-Benutzern verwendet werden.

Important

Dienstspezifische Anmeldeinformationen sind langfristige Anmeldeinformationen, die einem bestimmten IAM-Benutzer zugeordnet sind. Sie können nur für den Service verwendet werden, für den sie erstellt wurden. Um IAM-Rollen oder föderierten Identitäten Berechtigungen für den Zugriff auf all Ihre AWS Ressourcen mit temporären Anmeldeinformationen zu erteilen, sollten Sie die [AWS Authentifizierung mit dem SigV4-Authentifizierungs-Plugin](#) für Amazon Keyspaces verwenden.

Verwenden Sie eines der folgenden Verfahren, um dienstspezifische Anmeldeinformationen zu generieren.

Generieren Sie dienstspezifische Anmeldeinformationen mithilfe der Konsole

Um dienstspezifische Anmeldeinformationen mit der Konsole zu generieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Identity and Access Management Konsole unter <https://console.aws.amazon.com/iam/home>.

2. Wählen Sie im Navigationsbereich Benutzer und dann den Benutzer aus, den Sie zuvor erstellt haben und der über Amazon Keyspaces-Berechtigungen verfügt (Richtlinie beigefügt).
3. Wählen Sie Security Credentials (Sicherheitsanmeldeinformationen) aus. Wählen Sie unter Anmeldeinformationen für Amazon Keyspaces die Option Anmeldeinformationen generieren aus, um die dienstspezifischen Anmeldeinformationen zu generieren.


Ihre servicespezifischen Anmeldeinformationen sind nun verfügbar. Dies ist das einzige Mal, dass Sie das Passwort herunterladen oder anzeigen können. Später kann er nicht mehr wiederhergestellt werden. Sie können Ihr Passwort jedoch jederzeit zurücksetzen. Speichern Sie den Benutzer und das Passwort an einem sicheren Ort, da Sie sie später benötigen.

Generieren Sie dienstspezifische Anmeldeinformationen mit dem AWS CLI

Um dienstspezifische Anmeldeinformationen mit dem zu generieren AWS CLI

Bevor Sie dienstspezifische Anmeldeinformationen generieren, müssen Sie Folgendes herunterladen, installieren und konfigurieren AWS Command Line Interface :AWS CLI

1. Laden Sie das AWS CLI unter <http://aws.amazon.com/cli> herunter.

 Note

Das AWS CLI läuft unter Windows, MacOS oder Linux.

2. Folgen Sie den Anweisungen zur [Installation der AWS CLI](#) und [zur Konfiguration der AWS CLI](#) im AWS Command Line Interface Benutzerhandbuch.
3. Führen Sie mit dem folgenden Befehl den folgenden Befehl aus AWS CLI, um dienstspezifische Anmeldeinformationen für die Benutzerin zu generieren `alice`, damit sie auf Amazon Keyspaces zugreifen kann.


```
aws iam create-service-specific-credential \  
  --user-name alice \  
  --service-name cassandra.amazonaws.com
```

Die Ausgabe sieht wie folgt aus.

```
{  
  "ServiceSpecificCredential": {
```

```
"CreateDate": "2019-10-09T16:12:04Z",
"ServiceName": "cassandra.amazonaws.com",
"ServiceUserName": "alice-at-111122223333",
"ServicePassword": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
"ServiceSpecificCredentialId": "ACCAYFI33SINPGJEBYESF",
"UserName": "alice",
>Status": "Active"
}
}
```

Notieren Sie sich in der Ausgabe die Werte für `ServiceUserName` und `ServicePassword`. Speichern Sie diese Werte an einem sicheren Ort, da Sie sie später benötigen werden.

 **Important**

Dies ist das einzige Mal, dass `ServicePassword` Ihnen zur Verfügung steht.

So erstellen und konfigurieren Sie AWS Anmeldeinformationen für Amazon Keyspaces

Um programmgesteuert mit dem AWS SDK oder mit Cassandra-Client-Treibern und dem SigV4-Plugin auf Amazon Keyspaces zuzugreifen, benötigen Sie einen IAM-Benutzer oder eine IAM-Rolle mit Zugriffsschlüsseln. AWS CLI Wenn Sie AWS programmgesteuert verwenden, geben Sie Ihre AWS Zugriffsschlüssel an, damit Sie Ihre Identität bei programmatischen Aufrufen überprüfen können. AWS Ihre Zugangsschlüssel bestehen aus einer Zugriffsschlüssel-ID (zum Beispiel AKIAIOSFODNN7EXAMPLE) und einem geheimen Zugriffsschlüssel (zum Beispiel wjalrxUTNFEMI/K7MDENG/CYEXAMPLEKEY). bPxRfi In diesem Thema werden Sie durch die erforderlichen Schritte in diesem Prozess geführt.

Bewährte Sicherheitsmethoden empfehlen, IAM-Benutzer mit eingeschränkten Berechtigungen zu erstellen und stattdessen IAM-Rollen den Berechtigungen zuzuordnen, die für die Ausführung bestimmter Aufgaben erforderlich sind. IAM-Benutzer können dann vorübergehend IAM-Rollen übernehmen, um die erforderlichen Aufgaben auszuführen. Beispielsweise können IAM-Benutzer in Ihrem Konto, die die Amazon Keyspaces-Konsole verwenden, zu einer Rolle wechseln, um vorübergehend die Berechtigungen der Rolle in der Konsole zu nutzen. Die Benutzer verlieren ihre ursprünglichen Berechtigungen und übernehmen die Berechtigungen der zugewiesenen Rolle. Wenn der Benutzer die Rolle verlässt, werden die ursprünglichen Berechtigungen wiederhergestellt. Die Anmeldeinformationen, die die Benutzer verwenden, um die Rolle anzunehmen, sind temporär. Im Gegenteil, IAM-Benutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko

darstellt, wenn ihnen Rechte direkt zugewiesen werden, anstatt Rollen anzunehmen. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden. Weitere Informationen zu Rollen finden Sie im IAM-Benutzerhandbuch unter [Allgemeine Szenarien für Rollen: Benutzer, Anwendungen und Dienste](#).

Themen

- [Anmeldeinformationen AWS CLI, die für das AWS SDK oder das Amazon Keyspaces SigV4-Plugin für Cassandra-Client-Treiber erforderlich sind](#)
- [Einen IAM-Benutzer für den programmatischen Zugriff auf Amazon Keyspaces in Ihrem Konto erstellen AWS](#)
- [Neue Zugriffsschlüssel für einen IAM-Benutzer erstellen](#)
- [Wie verwaltet man Zugangsschlüssel für IAM-Benutzer](#)
- [Verwenden temporärer Anmeldeinformationen, um mithilfe einer IAM-Rolle und des Sigv4-Plug-ins eine Verbindung zu Amazon Keyspaces herzustellen](#)

Anmeldeinformationen AWS CLI, die für das AWS SDK oder das Amazon Keyspaces SigV4-Plugin für Cassandra-Client-Treiber erforderlich sind

Die folgenden Anmeldeinformationen sind erforderlich, um den IAM-Benutzer oder die IAM-Rolle zu authentifizieren:

AWS_ACCESS_KEY_ID

Gibt einen AWS Zugriffsschlüssel an, der einem IAM-Benutzer oder einer IAM-Rolle zugeordnet ist.

Der Zugriffsschlüssel `aws_access_key_id` ist erforderlich, um programmgesteuert eine Verbindung zu Amazon Keyspaces herzustellen.

AWS_SECRET_ACCESS_KEY

Gibt den geheimen Schlüssel an, der mit dem Zugriffsschlüssel verknüpft ist. Dies ist im Wesentlichen das "Passwort" für den Zugriffsschlüssel.

Das `aws_secret_access_key` ist erforderlich, um programmgesteuert eine Verbindung zu Amazon Keyspaces herzustellen.

AWS_SESSION_TOKEN— Fakultativ

Gibt den Sitzungstokenwert an, der erforderlich ist, wenn Sie temporäre Sicherheitsanmeldeinformationen verwenden, die Sie direkt aus AWS Security Token Service -Operationen abgerufen haben. Weitere Informationen finden Sie unter [the section called “Temporäre Anmeldeinformationen verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen”](#).

Wenn Sie eine Verbindung mit einem IAM-Benutzer herstellen, `aws_session_token` ist das nicht erforderlich.

Einen IAM-Benutzer für den programmatischen Zugriff auf Amazon Keyspaces in Ihrem Konto erstellen AWS

Um Anmeldeinformationen für den programmatischen Zugriff auf Amazon Keyspaces mit dem AWS CLI, dem AWS SDK oder dem SigV4-Plugin zu erhalten, müssen Sie zunächst einen IAM-Benutzer oder eine IAM-Rolle erstellen. Der Prozess der Erstellung eines IAM-Benutzers und der Konfiguration dieses IAM-Benutzers für den programmatischen Zugriff auf Amazon Keyspaces wird in den folgenden Schritten gezeigt:

1. Erstellen Sie den Benutzer in den AWS Management Console Tools für Windows oder mithilfe einer PowerShell API-Operation. AWS CLI AWS Wenn Sie den Benutzer in der erstellen AWS Management Console, werden die Anmeldeinformationen automatisch erstellt.
2. Wenn Sie den Benutzer programmgesteuert erstellen, müssen Sie in einem zusätzlichen Schritt einen Zugriffsschlüssel (Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel) für diesen Benutzer erstellen.
3. Erteilen Sie dem Benutzer Berechtigungen für den Zugriff auf Amazon Keyspaces.

Informationen zu den Berechtigungen, die Sie benötigen, um einen Benutzer zu erstellen, finden Sie unter [Erforderliche Berechtigungen für den Zugriff auf IAM-Ressourcen](#).


Erstellen von IAM-Benutzern (Konsole)

Sie können den verwenden AWS Management Console , um IAM-Benutzer zu erstellen.

Um einen IAM-Benutzer mit programmatischem Zugriff (Konsole) zu erstellen

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.](https://console.aws.amazon.com/iam/)

2. Wählen Sie im Navigationsbereich Users und dann Add User.
3. Geben Sie den Benutzernamen für den neuen Benutzer an. Dies ist der Anmeldename für AWS

 Note

Benutzernamen können eine Kombination aus bis zu 64 Buchstaben, Ziffern und den folgenden Zeichen enthalten: Plus (+), Gleichheitszeichen (=), Komma (,), Punkt (.), at-Zeichen (@), Unterstrich (_) und Bindestrich (-). Namen müssen innerhalb eines Kontos eindeutig sein. Es wird hierbei nicht zwischen Groß- und Kleinschreibung unterschieden. So können Sie beispielsweise keine zwei Gruppen mit Namen TESTBENUTZER und testbenutzer erstellen.

4. Wählen Sie Zugriffsschlüssel — Programmgesteuerter Zugriff aus, um einen Zugriffsschlüssel für den neuen Benutzer zu erstellen. Sie können den Zugriffsschlüssel ansehen oder herunterladen, wenn Sie zur letzten Seite gelangen.

Wählen Sie Weiter: Berechtigungen aus.

5. Wählen Sie auf der Seite Berechtigungen festlegen die Option Bestehende Richtlinien direkt anhängen aus, um dem neuen Benutzer Berechtigungen zuzuweisen.


Diese Option zeigt die Liste der AWS verwalteten und vom Kunden verwalteten Richtlinien an, die in Ihrem Konto verfügbar sind. Sie können keyspaces in das Suchfeld eingeben, um nur die Richtlinien anzuzeigen, die sich auf Amazon Keyspaces beziehen.

Für Amazon Keyspaces sind die verfügbaren verwalteten Richtlinien `AmazonKeyspacesFullAccess` und `AmazonKeyspacesReadOnlyAccess`. Weitere Informationen zu den einzelnen Richtlinien finden Sie unter [the section called “Von AWS verwaltete Richtlinien”](#).

Wählen Sie zu Testzwecken und um den Verbindungs-Tutorials zu folgen, die `AmazonKeyspacesReadOnlyAccess` Richtlinie für den neuen IAM-Benutzer aus. Hinweis: Als bewährte Methode empfehlen wir, das Prinzip der geringsten Rechte zu befolgen und benutzerdefinierte Richtlinien zu erstellen, die den Zugriff auf bestimmte Ressourcen einschränken und nur die erforderlichen Aktionen zulassen. Weitere Informationen zu IAM-Richtlinien und Beispielrichtlinien für Amazon Keyspaces finden Sie unter [the section called “Identitätsbasierte Richtlinien von Amazon Keyspaces”](#) Nachdem Sie benutzerdefinierte Berechtigungsrichtlinien erstellt haben, fügen Sie Ihre Richtlinien den Rollen hinzu und lassen Sie die Benutzer dann vorübergehend die entsprechenden Rollen übernehmen.

Wählen Sie Weiter: Markierungen.

6. Auf der Seite Stichwörter hinzufügen (optional) können Sie Stichwörter für den Benutzer hinzufügen oder Weiter: Überprüfen wählen.
7. Auf der Seite „Überprüfen“ können Sie alle Entscheidungen sehen, die Sie bis zu diesem Zeitpunkt getroffen haben. Wenn Sie bereit sind, fortzufahren, wählen Sie Benutzer erstellen aus.
8. Um die Zugriffsschlüssel des Benutzers (Zugriffsschlüssel-IDs und geheime Zugriffsschlüssel) anzuzeigen, wählen Sie neben dem Passwort und dem Zugriffsschlüssel die Option Show (Anzeigen) aus. Zum Speichern der Zugriffsschlüssel wählen Sie Download .csv. Speichern Sie die Datei an einem sicheren Speicherort.

 **Important**

Dies ist Ihre einzige Möglichkeit, die geheimen Zugriffsschlüssel anzusehen oder herunterzuladen, und Sie benötigen diese Informationen, bevor sie das SigV4-Plugin verwenden können. Speichern Sie die neue Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel des Benutzers an einem sicheren Speicherort. Sie haben nach diesem Schritt keinen Zugriff mehr auf die geheimen Zugriffsschlüssel.

Erstellen von IAM-Benutzern (AWS CLI)

Sie können das verwenden AWS CLI , um einen IAM-Benutzer zu erstellen.

Um einen IAM-Benutzer mit programmatischem Zugriff zu erstellen ()AWS CLI

1. Erstellen Sie einen Benutzer mit dem folgenden AWS CLI Code.
 - [aws iam create-user](#)
2. Erteilen Sie dem Benutzer programmatischen Zugriff. Dazu sind Zugriffsschlüssel erforderlich, die auf folgende Weise generiert werden können.
 - AWS CLI: [aws iam create-access-key](#)
 - Tools für Windows PowerShell: [New-IAMAccessKey](#)
 - IAM-API: [CreateAccessKey](#)

⚠ Important

Dies ist Ihre einzige Möglichkeit, die geheimen Zugriffsschlüssel anzusehen oder herunterzuladen, und Sie benötigen diese Informationen, bevor sie das SigV4-Plugin verwenden können. Speichern Sie die neue Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel des Benutzers an einem sicheren Speicherort. Sie haben nach diesem Schritt keinen Zugriff mehr auf die geheimen Zugriffsschlüssel.

3. Hängen Sie die `AmazonKeyspacesReadOnlyAccess` Richtlinie, die die Benutzerberechtigungen definiert, an den Benutzer an. Hinweis: Als bewährte Methode empfehlen wir, Benutzerberechtigungen zu verwalten, indem Sie den Benutzer zu einer Gruppe hinzufügen und der Gruppe eine Richtlinie anhängen, anstatt sie direkt an einen Benutzer anzuhängen.
 - AWS CLI: [aws iam attach-user-policy](#)

Neue Zugriffsschlüssel für einen IAM-Benutzer erstellen

Wenn Sie bereits einen IAM-Benutzer haben, können Sie jederzeit neue Zugriffsschlüssel erstellen. Weitere Informationen zur Schlüsselverwaltung, z. B. zur Rotation von Zugriffsschlüsseln, finden Sie unter [Zugriffsschlüssel für IAM-Benutzer verwalten](#).


So erstellen Sie Zugriffsschlüssel für einen IAM-Benutzer (Konsole)

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Klicken Sie im Navigationsbereich auf `Users` (Benutzer).
3. Wählen Sie den Namen des Benutzers, dessen Zugriffsschlüssel Sie erstellen möchten.
4. Wählen Sie auf der Übersichtsseite des Benutzers die Registerkarte `Sicherheitsanmeldeinformationen` aus.
5. Wählen Sie im Abschnitt `Access keys` (Zugriffsschlüssel) `Create access key` (Zugriffsschlüssel erstellen).

Wählen Sie zum Anzeigen des neuen Zugriffsschlüsselpaars `Show` (Anzeigen) aus. Ihre Anmeldeinformationen sehen etwa folgendermaßen aus:

- Zugriffsschlüssel-ID: `AKIAIOSFODNN7EXAMPLE`

- Geheimer Zugriffsschlüssel: bPxRfi wjalrxutnfeMI/K7MDENG/ CYEXAMPLEKEY

 Note

Sie haben keinen Zugriff auf den geheimen Zugriffsschlüssel mehr, nachdem das Dialogfeld geschlossen wird.

6. Wählen Sie zum Herunterladen des Schlüsselpaares Download .csv file aus. Speichern Sie die Schlüssel an einem sicheren Ort.
7. Nachdem Sie die CSV-Datei heruntergeladen haben, klicken Sie auf Close (Schließen).

Wenn Sie einen Zugriffsschlüssel erstellen, ist das Schlüsselpaar standardmäßig aktiv, und Sie können es sofort verwenden.

Wie verwaltet man Zugangsschlüssel für IAM-Benutzer

Als bewährte Methode empfehlen wir, Zugriffsschlüssel nicht direkt in Code einzubetten. Mit den AWS SDKs und den AWS Befehlszeilentools können Sie Zugriffsschlüssel an bekannten Orten platzieren, sodass Sie sie nicht im Code speichern müssen. Legen Sie Zugriffsschlüssel an einem der folgenden Orte ab:

- Umgebungsvariablen — Wählen Sie auf einem Mehrmandantensystem Benutzerumgebungsvariablen und nicht Systemumgebungsvariablen.
- CLI-Anmeldeinformationsdatei – Die `credentials`- und `config`-Dateien werden aktualisiert, wenn Sie den Befehl `aws configure` ausführen. Die `credentials` Datei befindet sich `~/.aws/credentials` unter Linux, macOS oder Unix oder unter Windows `C:\Users\USERNAME\.aws\credentials` unter. Diese Datei kann die Anmeldeinformationsdetails für das `default`-Profil und alle benannten Profile enthalten.
- CLI-Konfigurationsdatei – Die `credentials`- und `config`-Dateien werden aktualisiert, wenn Sie den Befehl `aws configure` ausführen. Die `config` Datei befindet sich `~/.aws/config` unter Linux, macOS oder Unix oder unter Windows `C:\Users\USERNAME\.aws\config` unter. Diese Datei enthält die Konfigurationseinstellungen für das Standardprofil sowie alle benannten Profile.

Das Speichern von Zugriffsschlüsseln als Umgebungsvariablen ist eine Voraussetzung für die [the section called “Authentifizierungs-Plugin für Java 4.x”](#) Der Client sucht mithilfe der standardmäßigen Anbieterkette für Anmeldeinformationen nach Anmeldeinformationen, und als Umgebungsvariablen

gespeicherte Zugriffsschlüssel haben Vorrang vor allen anderen Speicherorten, beispielsweise Konfigurationsdateien. Weitere Informationen finden Sie unter [Konfigurationseinstellungen und Rangfolge](#).

Die folgenden Beispiele zeigen, wie Sie Umgebungsvariablen für den Standardbenutzer konfigurieren können.

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of security token>
```

Durch die Festlegung der Umgebungsvariablen wird der verwendete Wert bis zum Ende der Shell-Sitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert. Sie können Variablen für zukünftige Sitzungen persistent machen, indem Sie sie im Startup-Skript Ihrer Shell festlegen.

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
C:\> setx AWS_SESSION_TOKEN AQoDYXdzEJr...<remainder of security token>
```

Bei Verwendung von [set](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert bis zum Ende der aktuellen Eingabeaufforderungssitzung oder bis zur Festlegung eines anderen Wertes für die Variable geändert. Bei Verwendung von [setx](#) zur Festlegung einer Umgebungsvariablen wird der verwendete Wert in der aktuellen Eingabeaufforderungssitzung und allen nach Ausführung des Befehls erstellten Eingabeaufforderungssitzungen geändert. Andere Befehls-Shells, die zum Zeitpunkt der Befehlsausführung bereits ausgeführt werden, sind hiervon nicht betroffen.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
PS C:\> $Env:AWS_SESSION_TOKEN="AQoDYXdzEJr...<remainder of security token>"
```

Wenn Sie an der PowerShell Eingabeaufforderung eine Umgebungsvariable festlegen, wie in den vorherigen Beispielen gezeigt, wird der Wert nur für die Dauer der aktuellen Sitzung gespeichert. Um die Einstellung der Umgebungsvariablen für alle Sitzungen PowerShell und

Befehlszeilensitzungen beizubehalten, speichern Sie sie mithilfe der Systemanwendung in der Systemsteuerung. Alternativ können Sie die Variable für alle future PowerShell Sitzungen festlegen, indem Sie sie zu Ihrem PowerShell Profil hinzufügen. Weitere Informationen zum Speichern von Umgebungsvariablen oder deren Beibehaltung über mehrere Sitzungen hinweg finden Sie in der [PowerShell Dokumentation](#).

Verwenden temporärer Anmeldeinformationen, um mithilfe einer IAM-Rolle und des Sigv4-Plug-ins eine Verbindung zu Amazon Keyspaces herzustellen

Für mehr Sicherheit können Sie [temporäre Anmeldeinformationen](#) verwenden, um sich mit dem SigV4-Plugin zu authentifizieren. In vielen Fällen benötigen Sie keine langfristigen Zugriffsschlüssel, die nie ablaufen (wie es bei IAM-Benutzern der Fall ist). Stattdessen können Sie eine IAM-Rolle erstellen und temporäre Sicherheitsanmeldedaten generieren. Temporäre Sicherheitsanmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel, enthalten aber auch ein Sicherheits-Token, das angibt, wann die Anmeldeinformationen ablaufen. Weitere Informationen zur Verwendung von IAM-Rollen anstelle von langfristigen Zugriffsschlüsseln finden Sie unter [Zu einer IAM-Rolle \(AWS API\) wechseln](#).

Um mit temporären Anmeldeinformationen zu beginnen, müssen Sie zunächst eine IAM-Rolle erstellen.

Erstellen Sie eine IAM-Rolle, die nur Lesezugriff auf Amazon Keyspaces gewährt

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
2. Wählen Sie im Navigationsbereich Rollen und anschließend Rolle erstellen aus.
3. Wählen Sie auf der Seite Rolle erstellen unter Typ der vertrauenswürdigen Entität auswählen die Option AWS Dienst aus. Wählen Sie unter Anwendungsfall auswählen die Option Amazon EC2 und dann Weiter aus.
4. Wählen Sie auf der Seite Berechtigungen hinzufügen unter Berechtigungsrichtlinien die Option Amazon Keyspaces Read Only Access aus der Richtlinienliste aus und klicken Sie dann auf Weiter.
5. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Namen für die Rolle ein und überprüfen Sie die Abschnitte Vertrauenswürdige Entitäten auswählen und Berechtigungen hinzufügen. Sie können auf dieser Seite auch optionale Tags für die Rolle hinzufügen. Wenn Sie fertig sind, wählen Sie Rolle erstellen aus. Merken Sie sich diesen Namen, da Sie ihn benötigen, wenn Sie Ihre Amazon EC2 EC2-Instance starten.

Um temporäre Sicherheitsanmeldeinformationen im Code zu verwenden, rufen Sie programmgesteuert eine AWS Security Token Service API auf `AssumeRole` und extrahieren die resultierenden Anmeldeinformationen und das Sitzungstoken aus Ihrer IAM-Rolle, die Sie im vorherigen Schritt erstellt haben. Anschließend verwenden Sie diese Werte als Anmeldeinformationen für nachfolgende Aufrufe von `AWSCassandraClient`. Das folgende Beispiel zeigt Pseudocode für die Verwendung temporärer Sicherheitsanmeldeinformationen:

```
assumeRoleResult = AssumeRole(role-arn);
tempCredentials = new SessionAWSCredentials(
    assumeRoleResult.AccessKeyId,
    assumeRoleResult.SecretAccessKey,
    assumeRoleResult.SessionToken);
cassandraRequest = CreateAmazoncassandraClient(tempCredentials);
```

Ein Beispiel, das temporäre Anmeldeinformationen mithilfe des Python-Treibers für den Zugriff auf Amazon Keyspaces implementiert, finden Sie unter [???](#).

Einzelheiten über den Aufruf von `AssumeRole`, `GetFederationToken` und anderen API-Vorgängen finden Sie in der [AWS Security Token Service -API-Referenz](#). Informationen dazu, wie die temporären Sicherheitsanmeldeinformationen und das Sitzungstoken aus dem Ergebnis abgerufen werden, finden Sie in der Dokumentation für das SDK, mit dem Sie arbeiten. Sie finden die Dokumentation für alle AWS SDKs auf der [AWS Hauptdokumentationsseite](#) im Abschnitt SDKs und Toolkits.

Service-Endpunkte für Amazon Keyspaces

Themen

- [Ports und Protokolle](#)
- [Globale Endpunkte](#)
- [AWS GovCloud \(US\) Region-FIPS-Endpunkte](#)
- [Endpunkte der chinesischen Regionen](#)

Ports und Protokolle

Sie können programmgesteuert auf Amazon Keyspaces zugreifen, indem Sie einen `cqlsh` Client ausführen, einen für Apache 2.0 lizenzierten Cassandra-Treiber verwenden oder das SDK und das AWS CLI SDK verwenden. AWS

Die folgende Tabelle zeigt die Ports und Protokolle für die verschiedenen Zugriffsmechanismen.

Programmatischer Zugriff	Port	Protocol (Protokoll)
CQLSH	9142	TLS
Cassandra-Treiber	9142	TLS
AWS CLI	443	HTTPS
AWS-SDK	443	HTTPS

Für TLS-Verbindungen verwendet Amazon Keyspaces die Starfield CA, um sich gegenüber dem Server zu authentifizieren. Weitere Informationen finden Sie unter [the section called “Wie konfiguriert man cqlsh Verbindungen für TLS manuell”](#) oder im Abschnitt [Bevor Sie beginnen zu Ihrem Treiber](#) im Kapitel. [the section called “Verwenden eines Cassandra-Client-Treibers”](#)

Globale Endpunkte

Amazon Keyspaces ist im Folgenden AWS-Regionen verfügbar. Diese Tabelle zeigt den verfügbaren Service-Endpunkt für jede Region.

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Ohio)	us-east-2	cassandra.us-east-2.amazonaws.com	HTTPS und TLS
USA Ost (Nord-Virginia)	us-east-1	cassandra.us-east-1.amazonaws.com cassandra-fips.us-east-1.amazonaws.com	HTTPS und TLS TLS
USA West (Nordkalifornien)	us-west-1	cassandra.us-west-1.amazonaws.com	HTTPS und TLS

Name der Region	Region	Endpoint	Protocol (Protokoll)
USA West (Oregon)	us-west-2	cassandra.us-west-2.amazonaws.com cassandra-fips.us-west-2.amazonaws.com	HTTPS und TLS TLS
Asien-Pazifik (Hongkong)	ap-east-1	cassandra.ap-east-1.amazonaws.com	HTTPS und TLS
Asien-Pazifik (Mumbai)	ap-south-1	cassandra.ap-south-1.amazonaws.com	HTTPS und TLS
Asien-Pazifik (Seoul)	ap-northeast-2	cassandra.ap-northeast-2.amazonaws.com	HTTPS und TLS
Asien-Pazifik (Singapur)	ap-southeast-1	cassandra.ap-southeast-1.amazonaws.com	HTTPS und TLS
Asien-Pazifik (Sydney)	ap-southeast-2	cassandra.ap-southeast-2.amazonaws.com	HTTPS und TLS
Asien-Pazifik (Tokio)	ap-northeast-1	cassandra.ap-northeast-1.amazonaws.com	HTTPS und TLS
Kanada (Zentral)	ca-central-1	cassandra.ca-central-1.amazonaws.com	HTTPS und TLS

Name der Region	Region	Endpoint	Protocol (Protokoll)
Europa (Frankfurt)	eu-central-1	cassandra.eu-central-1.amazonaws.com	HTTPS und TLS
Europa (Irland)	eu-west-1	cassandra.eu-west-1.amazonaws.com	HTTPS und TLS
Europa (London)	eu-west-2	cassandra.eu-west-2.amazonaws.com	HTTPS und TLS
Europa (Paris)	eu-west-3	cassandra.eu-west-3.amazonaws.com	HTTPS und TLS
Europa (Stockholm)	eu-north-1	cassandra.eu-north-1.amazonaws.com	HTTPS und TLS
Naher Osten (Bahrain)	me-south-1	cassandra.me-south-1.amazonaws.com	HTTPS und TLS
Südamerika (São Paulo)	sa-east-1	cassandra.sa-east-1.amazonaws.com	HTTPS und TLS
AWS GovCloud (US-Ost)	us-gov-east-1	cassandra.us-gov-east-1.amazonaws.com	HTTPS und TLS
AWS GovCloud (US-West)	us-gov-west-1	cassandra.us-gov-west-1.amazonaws.com	HTTPS und TLS

AWS GovCloud (US) Region-FIPS-Endpunkte

Verfügbare FIPS-Endpunkte in der. AWS GovCloud (US) Region Weitere Informationen finden Sie unter [Amazon Keyspaces im AWS GovCloud \(US\)Benutzerhandbuch](#).

Name der Region	Region	FIPS-Endpunkt	Protocol (Protokoll)
AWS GovCloud (US-Ost)	us-gov-east-1	cassandra.us-gov-east-1.amazonaws.com	HTTPS und TLS
AWS GovCloud (US-West)	us-gov-west-1	cassandra.us-gov-west-1.amazonaws.com	HTTPS und TLS

Endpunkte der chinesischen Regionen

Die folgenden Amazon Keyspaces-Endpunkte sind in den Regionen AWS Chinas verfügbar.

Um auf diese Endpunkte zugreifen zu können, müssen Sie sich für einen separaten Satz von Kontoanmeldeinformationen anmelden, die nur für die Regionen Chinas gelten. Weitere Informationen finden Sie unter [Registrierung, Konten und Anmeldeinformationen in China](#).

Name der Region	Region	Endpunkt	Protocol (Protokoll)
China (Peking)	cn-north-1	cassandra.cn-north-1.amazonaws.com.cn	HTTPS und TLS
China (Ningxia)	cn-northwest-1	cassandra.cn-northwest-1.amazonaws.com.cn	HTTPS und TLS

Verwenden **cqlsh**, um eine Verbindung zu Amazon Keyspaces herzustellen

Um eine Verbindung zu Amazon Keyspaces herzustellen **cqlsh**, können Sie den **cqlsh-expansion** verwenden. Dies ist ein Toolkit, das gängige Apache Cassandra-Tools wie **cqlsh**

und Hilfsprogramme enthält, die für Amazon Keyspaces vorkonfiguriert sind und gleichzeitig die volle Kompatibilität mit Apache Cassandra gewährleisten. Es `cqlsh-expansion` integriert das SigV4-Authentifizierungs-Plugin und ermöglicht es Ihnen, eine Verbindung mithilfe von IAM-Zugriffsschlüsseln anstelle von Benutzername und Passwort herzustellen. Sie müssen nur die `cqlsh` Skripts installieren, um eine Verbindung herzustellen, und nicht die vollständige Apache Cassandra-Distribution, da Amazon Keyspaces serverlos ist. Dieses einfache Installationspaket enthält die `cqlsh-expansion` und die klassischen `cqlsh` Skripte, die Sie auf jeder Plattform installieren können, die Python unterstützt.

Allgemeine Informationen dazu finden Sie `cqlsh` unter [cqlsh: Die CQL-Shell](#).

Themen

- [Verwenden von `cqlsh-expansion`, um eine Verbindung zu Amazon Keyspaces herzustellen](#)
- [Wie konfiguriert man `cqlsh` Verbindungen für TLS manuell](#)

Verwenden von `cqlsh-expansion`, um eine Verbindung zu Amazon Keyspaces herzustellen

Installation und Konfiguration des `cqlsh-expansion`

1. Um das `cqlsh-expansion` Python-Paket zu installieren, können Sie einen `pip` Befehl ausführen. Dadurch werden die `cqlsh-expansion` Skripts mithilfe einer Pip-Installation zusammen mit einer Datei, die eine Liste von Abhängigkeiten enthält, auf Ihrem Computer installiert. Das `--user` flag weist `pip` an, das Python-Benutzerinstallationsverzeichnis für Ihre Plattform zu verwenden. Auf einem Unix-basierten System sollte das das `~/.local/` Verzeichnis sein.

Sie benötigen Python 3, um das zu installieren `cqlsh-expansion`, um Ihre Python-Version herauszufinden, verwenden `Python --version`. Zur Installation können Sie den folgenden Befehl ausführen.

```
python3 -m pip install --user cqlsh-expansion
```

Die Ausgabe sollte in etwa so aussehen.

```
Collecting cqlsh-expansion
  Downloading cqlsh_expansion-0.9.6-py3-none-any.whl (153 kB)
##### 153.7/153.7 KB 3.3 MB/s eta 0:00:00
```

```

Collecting cassandra-driver
  Downloading cassandra_driver-3.28.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (19.1 MB)
##### 19.1/19.1 MB 44.5 MB/s eta 0:00:00
Requirement already satisfied: six>=1.12.0 in /usr/lib/python3/dist-packages (from
cqlsh-expansion) (1.16.0)
Collecting boto3
  Downloading boto3-1.29.2-py3-none-any.whl (135 kB)
##### 135.8/135.8 KB 17.2 MB/s eta 0:00:00
Collecting cassandra-sigv4>=4.0.2
  Downloading cassandra_sigv4-4.0.2-py2.py3-none-any.whl (9.8 kB)
Collecting botocore<1.33.0,>=1.32.2
  Downloading botocore-1.32.2-py3-none-any.whl (11.4 MB)
##### 11.4/11.4 MB 60.9 MB/s eta 0:00:00
Collecting s3transfer<0.8.0,>=0.7.0
  Downloading s3transfer-0.7.0-py3-none-any.whl (79 kB)
##### 79.8/79.8 KB 13.1 MB/s eta 0:00:00
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting geomet<0.3,>=0.1
  Downloading geomet-0.2.1.post1-py3-none-any.whl (18 kB)
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
##### 247.7/247.7 KB 33.1 MB/s eta 0:00:00
Requirement already satisfied: urllib3<2.1,>=1.25.4 in /usr/lib/python3/dist-
packages (from botocore<1.33.0,>=1.32.2->boto3->cqlsh-expansion) (1.26.5)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from
geomet<0.3,>=0.1->cassandra-driver->cqlsh-expansion) (8.0.3)
Installing collected packages: python-dateutil, jmespath, geomet, cassandra-driver,
botocore, s3transfer, boto3, cassandra-sigv4, cqlsh-expansion
  WARNING: The script geomet is installed in '/home/ubuntu/.local/bin' which is not
on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The scripts cqlsh, cqlsh-expansion and cqlsh-expansion.init are
installed in '/home/ubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
Successfully installed boto3-1.29.2 botocore-1.32.2 cassandra-driver-3.28.0
cassandra-sigv4-4.0.2 cqlsh-expansion-0.9.6 geomet-0.2.1.post1 jmespath-1.0.1
python-dateutil-2.8.2 s3transfer-0.7.0

```

Wenn sich das Installationsverzeichnis nicht im `PATH` befindet, müssen Sie es gemäß den Anweisungen Ihres Betriebssystems hinzufügen. Im Folgenden finden Sie ein Beispiel für Ubuntu Linux.

```
export PATH="$PATH:/home/ubuntu/.local/bin"
```

Um zu bestätigen, dass das Paket installiert ist, können Sie den folgenden Befehl ausführen.

```
cqlsh-expansion --version
```

Die Ausgabe sollte so aussehen.

```
cqlsh 6.1.0
```

- Um das zu konfigurieren `cqlsh-expansion`, können Sie nach der Installation ein Skript ausführen, um die folgenden Schritte automatisch auszuführen:
 - Erstellen Sie das `.cassandra` Verzeichnis im Home-Verzeichnis des Benutzers, falls es noch nicht vorhanden ist.
 - Kopieren Sie eine vorkonfigurierte `cqlshrc` Konfigurationsdatei in das `.cassandra` Verzeichnis.
 - Kopieren Sie das digitale Starfield-Zertifikat in das `.cassandra` Verzeichnis. Amazon Keyspaces verwendet dieses Zertifikat, um die sichere Verbindung mit Transport Layer Security (TLS) zu konfigurieren. Die Verschlüsselung bei der Übertragung bietet eine zusätzliche Datenschutzebene, indem Ihre Daten auf dem Weg zu und von Amazon Keyspaces verschlüsselt werden.

Um das Skript zunächst zu überprüfen, können Sie es im Github-Repo unter aufrufen. [post_install.py](#)

Um das Skript zu verwenden, können Sie den folgenden Befehl ausführen.

```
cqlsh-expansion.init
```


Note

Das Verzeichnis und die Datei, die durch das Post-Installationskript erstellt wurden `pip uninstall`, werden bei der Deinstallation `cqlsh-expansion` von `using` nicht entfernt und müssen manuell gelöscht werden.

Herstellen einer Verbindung zu Amazon Keyspaces mithilfe der `cqlsh-expansion`

1. Konfigurieren Sie Ihre AWS-Region und fügen Sie sie als Benutzerumgebungsvariable hinzu.

Um Ihre Standardregion als Umgebungsvariable auf einem Unix-basierten System hinzuzufügen, können Sie den folgenden Befehl ausführen. Für dieses Beispiel verwenden wir US East (Nord-Virginia).

```
export AWS_DEFAULT_REGION=us-east-1
```

Weitere Informationen zum Festlegen von Umgebungsvariablen, auch für andere Plattformen, finden Sie unter [So legen Sie Umgebungsvariablen](#) fest.

2. Finden Sie Ihren Service-Endpunkt.

Wählen Sie den entsprechenden Service-Endpunkt für Ihre Region aus. Informationen zu den verfügbaren Endpunkten für Amazon Keyspaces finden Sie unter [the section called "Service-Endpunkte"](#). In diesem Beispiel verwenden wir den Endpunkt `cassandra.us-east-1.amazonaws.com`.

3. Konfigurieren Sie die Authentifizierungsmethode.

Die empfohlene Methode zur Erhöhung der Sicherheit ist die Verbindung mit IAM-Zugriffsschlüsseln (IAM-Benutzer, Rollen und föderierte Identitäten).


Bevor Sie eine Verbindung mit IAM-Zugriffsschlüsseln herstellen können, müssen Sie die folgenden Schritte ausführen:

- a. Erstellen Sie einen IAM-Benutzer oder folgen Sie den bewährten Methoden und erstellen Sie eine IAM-Rolle, die IAM-Benutzer übernehmen können. Weitere Informationen zum Erstellen von IAM-Zugriffsschlüsseln finden Sie unter [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#).

- b. Erstellen Sie eine IAM-Richtlinie, die der Rolle (oder dem IAM-Benutzer) mindestens schreibgeschützten Zugriff auf Amazon Keyspaces gewährt. Weitere Informationen zu den Berechtigungen, die der IAM-Benutzer oder die IAM-Rolle benötigt, um eine Verbindung zu Amazon Keyspaces herzustellen, finden Sie unter [the section called “Zugreifen auf Amazon Keyspaces-Tabellen”](#)
- c. Fügen Sie die Zugriffsschlüssel des IAM-Benutzers zu den Umgebungsvariablen des Benutzers hinzu, wie im folgenden Beispiel gezeigt.

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJa1rXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
```

Weitere Informationen zum Festlegen von Umgebungsvariablen, auch für andere Plattformen, finden Sie unter [So legen Sie Umgebungsvariablen](#) fest.

 Note

Wenn Sie von einer Amazon EC2 EC2-Instance aus eine Verbindung herstellen, müssen Sie auch eine ausgehende Regel in der Sicherheitsgruppe konfigurieren, die den Datenverkehr von der Instance zu Amazon Keyspaces zulässt. Weitere Informationen zum Anzeigen und Bearbeiten von EC2-Regeln für ausgehenden Datenverkehr finden [Sie unter Regeln zu einer Sicherheitsgruppe hinzufügen im Amazon EC2 EC2-Benutzerhandbuch](#).

4. Stellen Sie mithilfe der `cqlsh-expansion` und SigV4-Authentifizierung eine Connect zu Amazon Keyspaces her.

Um mit dem eine Verbindung zu Amazon Keyspaces herzustellen `cqlsh-expansion`, können Sie den folgenden Befehl verwenden. Stellen Sie sicher, dass Sie den Service-Endpunkt durch den richtigen Endpunkt für Ihre Region ersetzen.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

Wenn die Verbindung erfolgreich ist, sollten Sie eine Ausgabe sehen, die dem folgenden Beispiel ähnelt.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
```

```
cqlsh current consistency level is ONE.  
cqlsh>
```

Wenn Sie auf einen Verbindungsfehler stoßen, finden Sie Informationen [the section called “Cqlsh-Verbindungsfehler”](#) zur Problembehandlung unter.

- Stellen Sie mit dienstspezifischen Anmeldeinformationen eine Connect zu Amazon Keyspaces her.

Um eine Verbindung mit der herkömmlichen Kombination aus Benutzername und Passwort herzustellen, die Cassandra für die Authentifizierung verwendet, müssen Sie zunächst dienstspezifische Anmeldeinformationen für Amazon Keyspaces erstellen, wie unter beschrieben. [the section called “Dienstspezifische Anmeldeinformationen”](#) Sie müssen diesem Benutzer auch Berechtigungen für den Zugriff auf Amazon Keyspaces erteilen. Weitere Informationen finden Sie unter [the section called “Zugreifen auf Amazon Keyspaces-Tabellen”](#).

Nachdem Sie dienstspezifische Anmeldeinformationen und Berechtigungen für den Benutzer erstellt haben, müssen Sie die `cqlshrc` Datei aktualisieren, die sich normalerweise im Benutzerverzeichnispfad befindet. `~/ .cassandra/` Gehen Sie in der `cqlshrc` Datei zum `[authentication]` Abschnitt Cassandra und kommentieren Sie das SigV4-Modul und die Klasse unter, `[auth_provider]` indem Sie das Zeichen „;“ verwenden, wie im folgenden Beispiel gezeigt.

```
[auth_provider]  
  
; module = cassandra_sigv4.auth  
  
; classname = SigV4AuthProvider
```

Nachdem Sie die `cqlshrc` Datei aktualisiert haben, können Sie mit dem folgenden Befehl eine Verbindung zu Amazon Keyspaces mit dienstspezifischen Anmeldeinformationen herstellen.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 -u myUserName -  
p myPassword --ssl
```

Bereinigen

- Um das `cqlsh-expansion` Paket zu entfernen, können Sie den `pip uninstall` Befehl verwenden.

```
pip3 uninstall cqlsh-expansion
```

Der `pip3 uninstall` Befehl entfernt nicht das Verzeichnis und die zugehörigen Dateien, die durch das Post-Installationsskript erstellt wurden. Um den Ordner und die Dateien zu entfernen, die durch das Post-Installationsskript erstellt wurden, können Sie das `.cassandra` Verzeichnis löschen.

Wie konfiguriert man **cqlsh** Verbindungen für TLS manuell

Amazon Keyspaces akzeptiert nur sichere Verbindungen mit Transport Layer Security (TLS). Sie können das `cqlsh-expansion` Hilfsprogramm verwenden, das das Zertifikat automatisch für Sie herunterlädt und eine vorkonfigurierte `cqlshrc` Konfigurationsdatei installiert. Weitere Informationen finden Sie [the section called “Verwenden des cqlsh-expansion”](#) auf dieser Seite.

Wenn Sie das Zertifikat herunterladen und die Verbindung manuell konfigurieren möchten, können Sie dies mit den folgenden Schritten tun.

1. Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secyreserver.net/repository/sf-class2-root.crt -O
```

Note

Sie können das digitale Amazon-Zertifikat auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen, und Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

2. Öffnen Sie beispielsweise die `cqlshrc` Konfigurationsdatei im Cassandra-Home-Verzeichnis `${HOME}/.cassandra/cqlshrc` und fügen Sie die folgenden Zeilen hinzu.

```
[connection]
```

```
port = 9142
factory = cqlshlib.ssl.ssl_transport_factory

[ssl]
validate = true
certfile = path_to_file/sf-class2-root.crt
```

Verwendung der AWS CLI

Das AWS Command Line Interface (AWS CLI) ermöglicht die Steuerung mehrerer AWS-Services über die Befehlszeile und automatisiert diese mithilfe von Skripten. Mit Amazon Keyspaces können Sie die AWS CLI Data Definition Language (DDL) -Operationen verwenden, wie etwa das Erstellen einer Tabelle. Darüber hinaus können Sie die Infrastructure-as-Code-Services (IaC) oder Terraform verwenden. AWS CloudFormation

Bevor Sie die AWS CLI mit Amazon Keyspaces verwenden können, müssen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel abrufen. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Eine vollständige Liste aller Befehle, die für Amazon Keyspaces in der verfügbar sind AWS CLI, finden Sie in der [AWS CLIBefehlszeilenreferenz](#).

Themen

- [Herunterladen und Konfigurieren der AWS CLI](#)
- [Die AWS CLI mit Amazon Keyspaces verwenden](#)

Herunterladen und Konfigurieren der AWS CLI

Das AWS CLI ist verfügbar unter <https://aws.amazon.com/cli>. Sie kann auf Windows, macOS oder Linux ausgeführt werden. Gehen Sie nach dem Herunterladen wie folgt vor AWS CLI, um es zu installieren und zu konfigurieren:

1. Gehe zum [AWS Command Line Interface Benutzerhandbuch](#)
2. Folgen Sie den Anweisungen [zur Installation AWS CLI](#) und [Konfiguration des AWS CLI](#)

Die AWS CLI mit Amazon Keyspaces verwenden

Das Befehlszeilenformat besteht aus einem Amazon Keyspaces-Operationsnamen gefolgt von den Parametern für diesen Vorgang. Die AWS CLI unterstützt eine Syntax-Kurznotation für die Parameterwerte und JSON. Die folgenden Amazon Keyspaces-Beispiele verwenden eine AWS CLI Abkürzungssyntax. Weitere Informationen finden Sie unter [Verwenden der Abkürzungssyntax mit der AWS CLI](#).

Mit dem folgenden Befehl wird ein Keyspace mit dem Namen `catalog` erstellt.

```
aws keyspaces create-keyspace --keyspace-name 'catalog'
```

Der Befehl gibt die Ressource Amazon Resource Name (ARN) in der Ausgabe zurück.

```
{
  "resourceArn": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/catalog/"
}
```

Um zu überprüfen, ob der Keyspace-Katalog vorhanden ist, können Sie den folgenden Befehl verwenden.

```
aws keyspaces get-keyspace --keyspace-name 'catalog'
```

Die Ausgabe des Befehls gibt die folgenden Werte zurück.

```
{
  "keyspaceName": "catalog",
  "resourceArn": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/catalog/"
}
```

Der folgende Befehl erstellt eine Tabelle mit dem Namen `book_awards`. Der Partitionsschlüssel der Tabelle besteht aus den Spalten `year` und `award` und der Clusterschlüssel besteht aus den Spalten `rank`, `category` und `author` und beide Clusterspalten verwenden die aufsteigende Sortierreihenfolge. (Für eine bessere Lesbarkeit werden lange Befehle in diesem Abschnitt über mehrere Zeilen verteilt.)

```
aws keyspaces create-table --keyspace-name 'catalog' --table-name 'book_awards'
  --schema-definition 'allColumns=[{name=year,type=int},
{name=award,type=text},{name=rank,type=int},
  {name=category,type=text}, {name=author,type=text},
{name=book_title,type=text},{name=publisher,type=text}]',
```

```
partitionKeys=[{name=year},
{name=award}],clusteringKeys=[{name=category,orderBy=ASC},{name=rank,orderBy=ASC}]'
```

Die Ausgabe dieses Befehls führt zu der folgenden Ausgabe.

```
{
  "resourceArn": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/catalog/table/
book_awards"
}
```

Um die Metadaten und Eigenschaften der Tabelle zu bestätigen, können Sie den folgenden Befehl verwenden.

```
aws keyspaces get-table --keyspace-name 'catalog' --table-name 'book_awards'
```

Dieser Befehl gibt die folgende Ausgabe zurück.

```
{
  "keyspaceName": "catalog",
  "tableName": "book_awards",
  "resourceArn": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/catalog/table/
book_awards",
  "creationTimestamp": 1645564368.628,
  "status": "ACTIVE",
  "schemaDefinition": {
    "allColumns": [
      {
        "name": "year",
        "type": "int"
      },
      {
        "name": "award",
        "type": "text"
      },
      {
        "name": "category",
        "type": "text"
      },
      {
        "name": "rank",
        "type": "int"
      }
    ]
  }
}
```

```
    {
      "name": "author",
      "type": "text"
    },
    {
      "name": "book_title",
      "type": "text"
    },
    {
      "name": "publisher",
      "type": "text"
    }
  ],
  "partitionKeys": [
    {
      "name": "year"
    },
    {
      "name": "award"
    }
  ],
  "clusteringKeys": [
    {
      "name": "category",
      "orderBy": "ASC"
    },
    {
      "name": "rank",
      "orderBy": "ASC"
    }
  ],
  "staticColumns": []
},
"capacitySpecification": {
  "throughputMode": "PAY_PER_REQUEST",
  "lastUpdateToPayPerRequestTimestamp": 1645564368.628
},
"encryptionSpecification": {
  "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
  "status": "DISABLED"
},
"ttl": {
```



```

    "status": "ENABLED"
  },
  "defaultTimeToLive": 0,
  "comment": {
    "message": ""
  }
}

```

Beim Erstellen von Tabellen mit komplexen Schemas kann es hilfreich sein, die Schemadefinition der Tabelle aus einer JSON-Datei zu laden. Es folgt ein Laden Sie die JSON-Beispieldatei für die Schemadefinition von [schema_definition.zip](#) herunter und extrahieren Sie `schemaschema_definition.json`, wobei Sie sich den Pfad zur Datei notieren. In diesem Beispiel für. Weitere Dateipfadoptionen finden Sie unter [So laden Sie Parameter aus einer Datei](#).

```

aws keyspaces create-table --keyspace-name 'catalog'
                        --table-name 'book_awards' --schema-definition 'file://
schema_definition.json'

```

Die folgenden Beispiele zeigen, wie Sie eine einfache Tabelle mit dem Namen MyTable mit zusätzlichen Optionen erstellen. Beachten Sie, dass die Befehle zur besseren Lesbarkeit in separate Zeilen aufgeteilt sind. Dieser Befehl zeigt, wie Sie eine -Tabelle erstellen und:

- stellen Sie den Kapazitätsmodus der Tabelle ein
- aktivieren Sie die `point-in-time P`-Wiederherstellung für die Tabelle
- setzt den TTL-Wert (Time to Live) für die -Tabelle auf ein Jahr
- füge zwei Tags für die Tabelle hinzu

```

aws keyspaces create-table --keyspace-name 'catalog' --table-name 'myTable'
                        --schema-definition 'allColumns=[{name=id,type=int},{name=name,type=text},
{name=date,type=timestamp}],partitionKeys=[{name=id}]'
                        --capacity-specification
'throughputMode=PROVISIONED,readCapacityUnits=5,writeCapacityUnits=5'
                        --point-in-time-recovery 'status=ENABLED'
                        --default-time-to-live '31536000'
                        --tags 'key=env,value=test' 'key=dpt,value=sec'

```

Dieses Beispiel zeigt, wie Sie eine neue Tabelle erstellen, die einen vom Kunden verwalteten Schlüssel für die Verschlüsselung verwendet und für die TTL aktiviert ist, sodass Sie Ablaufdaten für

Spalten und Zeilen festlegen können. Um dieses Beispiel auszuführen, müssen Sie den Ressourcen-ARN für den vom Kunden verwalteten AWS KMS Schlüssel durch Ihren eigenen Schlüssel ersetzen und sicherstellen, dass Amazon Keyspaces Zugriff darauf hat.

```
aws keyspaces create-table --keyspace-name 'catalog' --table-name 'myTable'
                        --schema-definition 'allColumns=[{name=id,type=int},{name=name,type=text},
                        {name=date,type=timestamp}],partitionKeys=[{name=id}]'
                        --encryption-specification
                        'type=CUSTOMER_MANAGED_KMS_KEY,kmsKeyIdentifier=arn:aws:kms:us-
                        east-1:111222333444:key/11111111-2222-3333-4444-555555555555'
                        --ttl 'status=ENABLED'
```

Verwenden der API

Sie können das AWS SDK und AWS Command Line Interface (AWS CLI) verwenden, um interaktiv mit Amazon Keyspaces zu arbeiten. Sie können die API für DDL-Operationen (Data Language Definition) verwenden, z. B. zum Erstellen eines Schlüsselraums oder einer Tabelle. Darüber hinaus können Sie Infrastructure as Code (IaC) -Dienste und Tools wie AWS CloudFormation Terraform verwenden.

Bevor Sie das AWS CLI mit Amazon Keyspaces verwenden können, benötigen Sie eine Zugangsschlüssel-ID und einen geheimen Zugriffsschlüssel. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Eine vollständige Liste aller Operationen, die für Amazon Keyspaces in der API verfügbar sind, finden Sie in der [Amazon Keyspaces API-Referenz](#).

Amazon Keyspaces mit einem AWS SDK verwenden

AWS Software Development Kits (SDKs) sind für viele gängige Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK for C++	AWS SDK for C++ Code-Beispiele
AWS CLI	AWS CLI Codebeispiele

SDK-Dokumentation	Codebeispiele
AWS SDK for Go	AWS SDK for Go Codebeispiele
AWS SDK for Java	AWS SDK for Java Codebeispiele
AWS SDK for JavaScript	AWS SDK for JavaScript Codebeispiele
AWS SDK for Kotlin	AWS SDK for Kotlin Codebeispiele
AWS SDK for .NET	AWS SDK for .NET Codebeispiele
AWS SDK for PHP	AWS SDK for PHP Codebeispiele
AWS Tools for PowerShell	Tools für PowerShell Codebeispiele
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) Codebeispiele
AWS SDK for Ruby	AWS SDK for Ruby Codebeispiele
AWS SDK for Rust	AWS SDK for Rust Codebeispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Codebeispiele
AWS SDK for Swift	AWS SDK for Swift Codebeispiele

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link [Provide feedback \(Feedback geben\)](#) auswählen.

Verwenden eines Cassandra-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

Sie können viele Open-Source-Cassandra-Treiber von Drittanbietern verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen. Amazon Keyspaces ist mit Cassandra-Treibern kompatibel, die Apache Cassandra Version 3.11.2 unterstützen. Dies sind die Treiber und neuesten Versionen, die wir getestet haben und die wir zur Verwendung mit Amazon Keyspaces empfehlen:

- Java v3.3
- Java v4.17
- Python Cassandra-driver 3.29.1
- Node.js cassandra driver -v 4.7.2
- GO using GOCQL v1.6
- .NET CassandraCSharpDriver -v 3.20.1

Weitere Informationen zu Cassandra-Treibern finden Sie unter [Apache Cassandra Client-Treiber](#).

Note

Um Ihnen den Einstieg zu erleichtern, können Sie end-to-end Codebeispiele ansehen und herunterladen, die Verbindungen zu Amazon Keyspaces mit gängigen Treibern herstellen.

[Beispiele für Amazon Keyspaces](#) finden Sie unter GitHub.

Die Tutorials in diesem Kapitel enthalten eine einfache CQL-Abfrage, um zu bestätigen, dass die Verbindung zu Amazon Keyspaces erfolgreich hergestellt wurde. Informationen zum Arbeiten mit Keyspaces und Tabellen, nachdem Sie eine Verbindung zu einem Amazon Keyspaces-Endpunkt hergestellt haben, finden Sie unter [CQL-Sprachreferenz](#). Ein step-by-step Tutorial, das zeigt, wie Sie von einem Amazon VPC-Endpunkt aus eine Verbindung zu Amazon Keyspaces herstellen, finden Sie unter [the section called “Verbindung mit VPC-Endpunkten herstellen”](#)

Themen

- [Verwenden eines Cassandra-Java-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)
- [Verwenden eines Cassandra-Python-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)
- [Verwenden eines Cassandra Node.js Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)
- [Verwenden eines Cassandra.NET-Core-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)
- [Verwenden eines Cassandra Go-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)

- [Verwenden eines Cassandra Perl-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces](#)

Verwenden eines Cassandra-Java-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

In diesem Abschnitt erfahren Sie, wie Sie mithilfe eines Java-Client-Treibers eine Verbindung zu Amazon Keyspaces herstellen.

Note

Java 17 und der DataStax Java-Treiber 4.17 werden derzeit nur in der Betaversion unterstützt. Weitere Informationen finden Sie unter https://docs.datastax.com/en/developer/java-driver/4.17/upgrade_guide/.

Um Benutzern und Anwendungen Anmeldeinformationen für den programmatischen Zugriff auf Amazon Keyspaces-Ressourcen zur Verfügung zu stellen, können Sie einen der folgenden Schritte ausführen:

- Erstellen Sie dienstspezifische Anmeldeinformationen, die einem bestimmten AWS Identity and Access Management (IAM-) Benutzer zugeordnet sind.
- Aus Sicherheitsgründen empfehlen wir, IAM-Zugriffsschlüssel für IAM-Identitäten zu erstellen, die für alle Dienste verwendet werden. AWS Das Amazon Keyspaces SigV4-Authentifizierungs-Plugin für Cassandra-Client-Treiber ermöglicht es Ihnen, Anrufe an Amazon Keyspaces mithilfe von IAM-Zugriffsschlüsseln anstelle von Benutzername und Passwort zu authentifizieren. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Note

Ein Beispiel für die Verwendung von Amazon Keyspaces mit Spring Boot finden Sie unter <https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/spring>.

Themen

- [Bevor Sie beginnen](#)
- [Ein tep-by-step Tutorial zum Herstellen einer Verbindung zu Amazon Keyspaces mithilfe des DataStax Java-Treibers für Apache Cassandra mit dienstspezifischen Anmeldeinformationen](#)
- [Ein tep-by-step Tutorial zum Herstellen einer Verbindung zu Amazon Keyspaces mithilfe des DataStax 4.x-Java-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins](#)
- [Stellen Sie mithilfe des DataStax 3.x-Java-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her](#)

Bevor Sie beginnen

Um eine Verbindung zu Amazon Keyspaces herzustellen, müssen Sie die folgenden Aufgaben ausführen, bevor Sie beginnen können.

1. Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern.
 - a. Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -0
```

Note

Sie können das digitale Amazon-Zertifikat auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen, und Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

- b. Konvertieren Sie das digitale Starfield-Zertifikat in eine TrustStore-Datei.

```
openssl x509 -outform der -in sf-class2-root.crt -out temp_file.der  
keytool -import -alias cassandra -keystore cassandra_truststore.jks -file  
temp_file.der
```

In diesem Schritt müssen Sie ein Passwort für den Keystore erstellen und diesem Zertifikat vertrauen. Der interaktive Befehl sieht so aus.

```

Enter keystore password:
Re-enter new password:
Owner: OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
  Inc.", C=US
Issuer: OU=Starfield Class 2 Certification Authority, O="Starfield
  Technologies, Inc.", C=US
Serial number: 0
Valid from: Tue Jun 29 17:39:16 UTC 2004 until: Thu Jun 29 17:39:16 UTC 2034
Certificate fingerprints:
  MD5: 32:4A:4B:BB:C8:63:69:9B:BE:74:9A:C6:DD:1D:46:24
  SHA1: AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A
  SHA256:
  14:65:FA:20:53:97:B8:76:FA:A6:F0:A9:95:8E:55:90:E4:0F:CC:7F:AA:4F:B7:C2:C8:67:75:21:FB
Signature algorithm name: SHA1withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3
Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....
]
[OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
  Inc.", C=US]
SerialNumber: [ 00]
]
#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....
]
]
Trust this certificate? [no]: y

```

2. Hängen Sie die TrustStore-Datei an die JVM-Argumente an:

```
-Djavax.net.ssl.trustStore=path_to_file/cassandra_truststore.jks  
-Djavax.net.ssl.trustStorePassword=my_password
```

Ein tep-by-step Tutorial zum Herstellen einer Verbindung zu Amazon Keyspaces mithilfe des DataStax Java-Treibers für Apache Cassandra mit dienstspezifischen Anmeldeinformationen

Das folgende step-by-step Tutorial führt Sie durch die Herstellung einer Verbindung zu Amazon Keyspaces mithilfe eines Java-Treibers für Cassandra mit dienstspezifischen Anmeldeinformationen. Insbesondere verwenden Sie die 4.0-Version des DataStax Java-Treibers für Apache Cassandra.

Themen

- [Schritt 1: Voraussetzungen](#)
- [Schritt 2: Konfigurieren Sie den Treiber](#)
- [Schritt 3: Führen Sie die Beispielanwendung aus](#)

Schritt 1: Voraussetzungen

Um diesem Tutorial zu folgen, müssen Sie dienstspezifische Anmeldeinformationen generieren und den DataStax Java-Treiber für Apache Cassandra zu Ihrem Java-Projekt hinzufügen.

- Generieren Sie dienstspezifische Anmeldeinformationen für Ihren Amazon Keyspaces IAM-Benutzer, indem Sie die unter aufgeführten Schritte ausführen. [the section called “Dienstspezifische Anmeldeinformationen”](#) Wenn Sie lieber IAM-Zugriffsschlüssel für die Authentifizierung verwenden möchten, finden Sie weitere Informationen unter. [the section called “Authentifizierungs-Plugin für Java 4.x”](#)
- Fügen Sie den DataStax Java-Treiber für Apache Cassandra zu Ihrem Java-Projekt hinzu. Stellen Sie sicher, dass Sie eine Version des Treibers verwenden, die Apache Cassandra 3.11.2 unterstützt. Weitere Informationen finden Sie in der Dokumentation zum [DataStax Java-Treiber für Apache Cassandra](#).


Schritt 2: Konfigurieren Sie den Treiber

Sie können Einstellungen für den DataStax Java-Cassandra-Treiber angeben, indem Sie eine Konfigurationsdatei für Ihre Anwendung erstellen. Diese Konfigurationsdatei überschreibt die Standardeinstellungen und weist den Treiber an, über Port 9142 eine Verbindung zum Amazon

Keyspaces-Serviceendpunkt herzustellen. Eine Liste der verfügbaren Service-Endpunkte finden Sie unter [the section called "Service-Endpunkte"](#)

Erstellen Sie eine Konfigurationsdatei und speichern Sie die Datei im Ressourcenordner der Anwendung, z. B. `src/main/resources/application.conf`. Öffnen Sie die folgenden `application.conf` Konfigurationseinstellungen und fügen Sie sie hinzu.

1. Authentifizierungsanbieter — Erstellen Sie den Authentifizierungsanbieter mit der `PlainTextAuthProvider` Klasse. `ServiceUserName` und `ServicePassword` sollten mit dem Benutzernamen und dem Passwort übereinstimmen, die Sie bei der Generierung der dienstspezifischen Anmeldeinformationen erhalten haben, indem Sie die Schritte unter ausführen. [Generieren Sie dienstspezifische Anmeldeinformationen](#)

 Note

Sie können kurzfristige Anmeldeinformationen verwenden, indem Sie das Authentifizierungs-Plug-In für den DataStax Java-Treiber für Apache Cassandra verwenden, anstatt die Anmeldeinformationen in Ihrer Treiberkonfigurationsdatei fest zu codieren. Um mehr zu erfahren, folgen Sie den Anweisungen für [the section called "Authentifizierungs-Plugin für Java 4.x"](#)

2. Lokales Rechenzentrum — Stellen Sie den Wert für `local-datacenter` die Region ein, mit der Sie eine Verbindung herstellen. Wenn die Anwendung beispielsweise eine Verbindung herstellt `cassandra.us-east-2.amazonaws.com`, legen Sie das lokale Rechenzentrum auf `festus-east-2`. Alle verfügbaren AWS-Regionen Informationen finden Sie unter [???](#). Auf `slow-replica-avoidance = false` Lastenausgleich für weniger Knoten eingestellt.
3. SSL/TLS — Initialisieren Sie SSL, EngineFactory indem Sie der Konfigurationsdatei einen Abschnitt mit einer einzigen Zeile hinzufügen, in der die Klasse mit angegeben wird. `class = DefaultSslEngineFactory` Geben Sie den Pfad zur TrustStore-Datei und das Passwort an, das Sie zuvor erstellt haben. Amazon Keyspaces unterstützt keine `hostname-validation` Peers, setzen Sie diese Option daher auf `False`.

```
datastax-java-driver {  
  
    basic.contact-points = [ "cassandra.us-east-2.amazonaws.com:9142"  
    advanced.auth-provider{  
        class = PlainTextAuthProvider  
        username = "ServiceUserName"
```

```
    password = "ServicePassword"
  }
  basic.load-balancing-policy {
    local-datacenter = "us-east-2"
    slow-replica-avoidance = false
  }

  advanced.ssl-engine-factory {
    class = DefaultSslEngineFactory
    truststore-path = "./src/main/resources/cassandra_truststore.jks"
    truststore-password = "my_password"
    hostname-validation = false
  }
}
```

Note

Anstatt den Pfad zum TrustStore in der Konfigurationsdatei hinzuzufügen, können Sie den TrustStore-Pfad auch direkt im Anwendungscode hinzufügen oder Sie können den Pfad zum TrustStore zu Ihren JVM-Argumenten hinzufügen.

Schritt 3: Führen Sie die Beispielanwendung aus

Dieses Codebeispiel zeigt eine einfache Befehlszeilenanwendung, die mithilfe der zuvor erstellten Konfigurationsdatei einen Verbindungspool zu Amazon Keyspaces erstellt. Es bestätigt, dass die Verbindung hergestellt wurde, indem eine einfache Abfrage ausgeführt wird.

```
package <your package>;
// add the following imports to your project
import com.datastax.oss.driver.api.core.CqlSession;
import com.datastax.oss.driver.api.core.config.DriverConfigLoader;
import com.datastax.oss.driver.api.core.cql.ResultSet;
import com.datastax.oss.driver.api.core.cql.Row;

public class App
{

    public static void main( String[] args )
    {
        //Use DriverConfigLoader to load your configuration file
```

```
DriverConfigLoader loader =
DriverConfigLoader.fromClasspath("application.conf");
try (CqlSession session = CqlSession.builder()
    .withConfigLoader(loader)
    .build()) {

    ResultSet rs = session.execute("select * from system_schema.keyspaces");
    Row row = rs.one();
    System.out.println(row.getString("keyspace_name"));
}
}
```

Note

Verwenden Sie einen `try` Block, um die Verbindung herzustellen, um sicherzustellen, dass sie immer geschlossen ist. Wenn Sie keinen `try` Block verwenden, denken Sie daran, Ihre Verbindung zu schließen, um zu vermeiden, dass Ressourcen verloren gehen.

Ein `tep-by-step` Tutorial zum Herstellen einer Verbindung zu Amazon Keyspaces mithilfe des DataStax 4.x-Java-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins

Im folgenden Abschnitt wird beschrieben, wie Sie das SigV4-Authentifizierungs-Plugin für den DataStax Open-Source-4.x-Java-Treiber für Apache Cassandra verwenden, um auf Amazon Keyspaces (für Apache Cassandra) zuzugreifen. [Das Plugin ist im Repository verfügbar. GitHub](#)

Das SigV4-Authentifizierungs-Plugin ermöglicht es Ihnen, IAM-Anmeldeinformationen für Benutzer oder Rollen zu verwenden, wenn Sie eine Verbindung zu Amazon Keyspaces herstellen. Anstatt einen Benutzernamen und ein Passwort zu benötigen, signiert dieses Plugin API-Anfragen mit Zugriffsschlüsseln. Weitere Informationen finden Sie unter [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#).


Schritt 1: Voraussetzungen

Um diesem Tutorial zu folgen, müssen Sie die folgenden Aufgaben ausführen.

- Falls Sie dies noch nicht getan haben, erstellen Sie Anmeldeinformationen für Ihren IAM-Benutzer oder Ihre IAM-Rolle gemäß den Schritten unter [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#). In dieser Anleitung wird davon ausgegangen, dass die

Zugriffsschlüssel als Umgebungsvariablen gespeichert sind. Weitere Informationen finden Sie unter [the section called “Wie verwaltet man Zugangsschlüssel”](#).

- Fügen Sie den DataStax Java-Treiber für Apache Cassandra zu Ihrem Java-Projekt hinzu. Stellen Sie sicher, dass Sie eine Version des Treibers verwenden, die Apache Cassandra 3.11.2 unterstützt. Weitere Informationen finden Sie in der Dokumentation zum [DataStax Java-Treiber für Apache Cassandra](#).
- Fügen Sie das Authentifizierungs-Plugin zu Ihrer Anwendung hinzu. Das Authentifizierungs-Plugin unterstützt Version 4.x des DataStax Java-Treibers für Apache Cassandra. Wenn Sie Apache Maven oder ein Build-System verwenden, das Maven-Abhängigkeiten verwenden kann, fügen Sie Ihrer Datei die folgenden Abhängigkeiten hinzu. `pom.xml`

 **Important**

[Ersetzen Sie die Version des Plugins durch die neueste Version, wie im GitHub Repository gezeigt.](#)

```
<dependency>
  <groupId>software.aws.mcs</groupId>
  <artifactId>aws-sigv4-auth-cassandra-java-driver-plugin</artifactId>
  <version>4.0.9</version>
</dependency>
```

Schritt 2: Konfigurieren Sie den Treiber

Sie können Einstellungen für den DataStax Java-Cassandra-Treiber angeben, indem Sie eine Konfigurationsdatei für Ihre Anwendung erstellen. Diese Konfigurationsdatei überschreibt die Standardeinstellungen und weist den Treiber an, über Port 9142 eine Verbindung zum Amazon Keyspaces-Serviceendpunkt herzustellen. Eine Liste der verfügbaren Service-Endpunkte finden Sie unter [the section called “Service-Endpunkte”](#)

Erstellen Sie eine Konfigurationsdatei und speichern Sie die Datei im Ressourcenordner der Anwendung, z. B. `src/main/resources/application.conf`. Öffnen Sie die folgenden `application.conf` Konfigurationseinstellungen und fügen Sie sie hinzu.

1. **Authentifizierungsanbieter** — Legen Sie `advanced.auth-provider.class` für eine neue Instanz von `festsoftware.aws.mcs.auth.SigV4AuthProvider`. Der SigV4 AuthProvider ist

- der Authentifizierungshandler, der vom Plugin für die Durchführung der SigV4-Authentifizierung bereitgestellt wird.
2. Lokales Rechenzentrum — Stellen Sie den Wert für die Region ein `local-datacenter`, zu der Sie eine Verbindung herstellen. Wenn die Anwendung beispielsweise eine Verbindung herstellt `cassandra.us-east-2.amazonaws.com`, legen Sie das lokale Rechenzentrum auf `festus-east-2`. Alle verfügbaren AWS-Regionen Informationen finden Sie unter [???](#). Stellt `slow-replica-avoidance = false` den Lastenausgleich für alle verfügbaren Knoten ein.
 3. Idempotenz — Legen Sie als Standardeinstellung `idempotence` für die Anwendung fest, dass der Treiber so konfiguriert wird, dass `true` fehlgeschlagene Anfragen beim Lesen/Schreiben/Vorbereiten/Ausführen immer wiederholt werden. Dies ist eine bewährte Methode für verteilte Anwendungen, mit der vorübergehende Fehler behoben werden können, indem fehlgeschlagene Anfragen wiederholt werden.
 4. SSL/TLS — Initialisieren Sie SSL, EngineFactory indem Sie der Konfigurationsdatei einen Abschnitt mit einer einzigen Zeile hinzufügen, in der die Klasse mit angegeben wird. `class = DefaultSslEngineFactory` Geben Sie den Pfad zur TrustStore-Datei und das Passwort an, das Sie zuvor erstellt haben. Amazon Keyspaces unterstützt keine `hostname-validation` Peers, setzen Sie diese Option daher auf `False`.
 5. Verbindungen — Erstellen Sie mindestens 3 lokale Verbindungen pro Endpunkt, indem Sie diese Einstellung `local.size = 3` vornehmen. Dies ist eine bewährte Methode, die Ihrer Anwendung hilft, Overhead und Datenverkehrsspitzen zu bewältigen. Weitere Informationen darüber, wie Sie anhand der erwarteten Datenverkehrsmuster berechnen können, wie viele lokale Verbindungen pro Endpunkt Ihre Anwendung benötigt, finden Sie unter [the section called "Wie konfiguriert man Verbindungen"](#).
 6. Wiederholungsrichtlinie — Die Wiederholungsrichtlinie von Amazon Keyspaces `AmazonKeyspacesExponentialRetryPolicy` ist eine Alternative zu der Richtlinie `DefaultRetryPolicy`, die im Cassandra-Treiber enthalten ist. Der Hauptunterschied zwischen den beiden Wiederholungsrichtlinien besteht darin, dass Sie die Anzahl der Wiederholungsversuche so konfigurieren können, dass sie Ihren Anforderungen entspricht. `AmazonKeyspacesExponentialRetryPolicy` Standardmäßig ist die Anzahl der Wiederholungsversuche für auf 3 `AmazonKeyspacesExponentialRetryPolicy` festgelegt. Darüber hinaus gibt die Amazon Keyspaces-Wiederholungsrichtlinie das Generikum nicht zurück. `NoHostAvailableException` Stattdessen gibt die Amazon Keyspaces-Wiederholungsrichtlinie die ursprüngliche Ausnahme zurück, die vom Service zurückgegeben wurde. Weitere Codebeispiele zur Implementierung von Wiederholungsrichtlinien finden Sie unter [Amazon Keyspaces-Wiederholungsrichtlinien](#) auf Github.

7. Vorbereitete Anweisungen — Auf „False“ setzen `prepare-on-all-nodes`, um die Netzwerknutzung zu optimieren.

```
datastax-java-driver {
  basic {
    contact-points = [ "cassandra.us-east-2.amazonaws.com:9142" ]
    request {
      timeout = 2 seconds
      consistency = LOCAL_QUORUM
      page-size = 1024
      default-idempotence = true
    }
    load-balancing-policy {
      local-datacenter = "us-east-2"
      class = DefaultLoadBalancingPolicy
      slow-replica-avoidance = false
    }
  }
  advanced {
    auth-provider {
      class = software.aws.mcs.auth.SigV4AuthProvider
      aws-region = us-east-2
    }
    ssl-engine-factory {
      class = DefaultSslEngineFactory
      truststore-path = "./src/main/resources/cassandra_truststore.jks"
      truststore-password = "my_password"
      hostname-validation = false
    }
    connection {
      connect-timeout = 5 seconds
      max-requests-per-connection = 512
      pool {
        local.size = 3
      }
    }
    retry-policy {
      class = com.aws.ssa.keyspaces.retry.AmazonKeyspacesExponentialRetryPolicy
    }
    max-attempts = 3
    min-wait = 10 mills
    max-wait = 100 mills
  }
}
```

```
    prepared-statements {
      prepare-on-all-nodes = false
    }
  }
}
```

Note

Anstatt den Pfad zum TrustStore in der Konfigurationsdatei hinzuzufügen, können Sie den TrustStore-Pfad auch direkt im Anwendungscode hinzufügen oder Sie können den Pfad zum TrustStore zu Ihren JVM-Argumenten hinzufügen.

Schritt 3: Führen Sie die Anwendung aus

Dieses Codebeispiel zeigt eine einfache Befehlszeilenanwendung, die mithilfe der zuvor erstellten Konfigurationsdatei einen Verbindungspool zu Amazon Keyspaces erstellt. Es bestätigt, dass die Verbindung hergestellt wurde, indem eine einfache Abfrage ausgeführt wird.

```
package <your package>;
// add the following imports to your project
import com.datastax.oss.driver.api.core.CqlSession;
import com.datastax.oss.driver.api.core.config.DriverConfigLoader;
import com.datastax.oss.driver.api.core.cql.ResultSet;
import com.datastax.oss.driver.api.core.cql.Row;

public class App
{

    public static void main( String[] args )
    {
        //Use DriverConfigLoader to load your configuration file
        DriverConfigLoader loader =
        DriverConfigLoader.fromClasspath("application.conf");
        try (CqlSession session = CqlSession.builder()
            .withConfigLoader(loader)
            .build()) {

            ResultSet rs = session.execute("select * from system_schema.keyspaces");
            Row row = rs.one();
            System.out.println(row.getString("keyspace_name"));

        }
    }
}
```

```
}  
}
```

Note

Verwenden Sie einen `try` Block, um die Verbindung herzustellen, um sicherzustellen, dass sie immer geschlossen ist. Wenn Sie keinen `try` Block verwenden, denken Sie daran, Ihre Verbindung zu schließen, um zu vermeiden, dass Ressourcen verloren gehen.

Stellen Sie mithilfe des DataStax 3.x-Java-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her

Im folgenden Abschnitt wird beschrieben, wie Sie das SigV4-Authentifizierungs-Plugin für den DataStax 3.x-Open-Source-Java-Treiber für Apache Cassandra für den Zugriff auf Amazon Keyspaces verwenden. [Das Plugin ist im Repository verfügbar. GitHub](#)

Das SigV4-Authentifizierungs-Plugin ermöglicht es Ihnen, IAM-Anmeldeinformationen für Benutzer und Rollen zu verwenden, wenn Sie eine Verbindung zu Amazon Keyspaces herstellen. Anstatt einen Benutzernamen und ein Passwort zu benötigen, signiert dieses Plugin API-Anfragen mit Zugriffsschlüsseln. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Schritt 1: Voraussetzungen

Um dieses Codebeispiel auszuführen, müssen Sie zunächst die folgenden Aufgaben ausführen.

- Erstellen Sie Anmeldeinformationen für Ihren IAM-Benutzer oder Ihre IAM-Rolle, indem Sie die Schritte unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#) befolgen. In dieser Anleitung wird davon ausgegangen, dass die Zugriffsschlüssel als Umgebungsvariablen gespeichert sind. Weitere Informationen finden Sie unter [the section called “Wie verwaltet man Zugangsschlüssel”](#).
- Gehen Sie wie unter beschrieben vor, [the section called “Bevor Sie beginnen”](#) um das digitale Starfield-Zertifikat herunterzuladen, es in eine TrustStore-Datei zu konvertieren und die TrustStore-Datei in den JVM-Argumenten an Ihre Anwendung anzuhängen.
- Fügen Sie den DataStax Java-Treiber für Apache Cassandra zu Ihrem Java-Projekt hinzu. Stellen Sie sicher, dass Sie eine Version des Treibers verwenden, die Apache Cassandra 3.11.2 unterstützt. Weitere Informationen finden Sie in der Dokumentation zum [DataStax Java-Treiber für Apache Cassandra](#).

- Fügen Sie das Authentifizierungs-Plugin zu Ihrer Anwendung hinzu. Das Authentifizierungs-Plugin unterstützt Version 3.x des DataStax Java-Treibers für Apache Cassandra. Wenn Sie Apache Maven oder ein Build-System verwenden, das Maven-Abhängigkeiten verwenden kann, fügen Sie Ihrer Datei die folgenden Abhängigkeiten hinzu. pom.xml [Ersetzen Sie die Version des Plugins durch die neueste Version, wie im GitHub Repository gezeigt.](#)

```
<dependency>
  <groupId>software.aws.mcs</groupId>
  <artifactId>aws-sigv4-auth-cassandra-java-driver-plugin_3</artifactId>
  <version>3.0.3</version>
</dependency>
```

Schritt 2: Führen Sie die Anwendung aus

Dieses Codebeispiel zeigt eine einfache Befehlszeilenanwendung, die einen Verbindungspool zu Amazon Keyspaces erstellt. Es bestätigt, dass die Verbindung hergestellt wurde, indem eine einfache Abfrage ausgeführt wird.

```
package <your package>;
// add the following imports to your project

import software.aws.mcs.auth.SigV4AuthProvider;
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.ResultSet;
import com.datastax.driver.core.Row;
import com.datastax.driver.core.Session;

public class App
{

    public static void main( String[] args )
    {
        String endPoint = "cassandra.us-east-2.amazonaws.com";
        int portNumber = 9142;
        Session session = Cluster.builder()
            .addContactPoint(endPoint)
            .withPort(portNumber)
            .withAuthProvider(new SigV4AuthProvider("us-east-2"))

            .withSSL()
            .build()
```

```
        .connect();

        ResultSet rs = session.execute("select * from system_schema.keyspaces");
        Row row = rs.one();
        System.out.println(row.getString("keyspace_name"));
    }
}
```

Hinweise zur Verwendung:

Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called “Service-Endpunkte”](#).

Im folgenden Repository finden Sie hilfreiche Java-Treiberrichtlinien, Beispiele und bewährte Methoden für die Verwendung des Java-Treibers mit Amazon Keyspaces: <https://github.com/aws-samples/amazon-keyspaces-java-driver-helpers>.

Verwenden eines Cassandra-Python-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

In diesem Abschnitt zeigen wir Ihnen, wie Sie mithilfe eines Python-Client-Treibers eine Verbindung zu Amazon Keyspaces herstellen. Um Benutzern und Anwendungen Anmeldeinformationen für den programmatischen Zugriff auf Amazon Keyspaces-Ressourcen zur Verfügung zu stellen, können Sie einen der folgenden Schritte ausführen:

- Erstellen Sie dienstspezifische Anmeldeinformationen, die einem bestimmten AWS Identity and Access Management (IAM-) Benutzer zugeordnet sind.
- Aus Sicherheitsgründen empfehlen wir, IAM-Zugriffsschlüssel für IAM-Benutzer oder -Rollen zu erstellen, die für alle Dienste verwendet werden. AWS Das Amazon Keyspaces SigV4-Authentifizierungs-Plugin für Cassandra-Client-Treiber ermöglicht es Ihnen, Anrufe an Amazon Keyspaces mithilfe von IAM-Zugriffsschlüsseln anstelle von Benutzername und Passwort zu authentifizieren. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Themen

- [Bevor Sie beginnen](#)
- [Stellen Sie mithilfe des Python-Treibers für Apache Cassandra und dienstspezifische Anmeldeinformationen eine Connect zu Amazon Keyspaces her](#)

- [Stellen Sie mithilfe des DataStax Python-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her](#)

Bevor Sie beginnen

Sie müssen die folgende Aufgabe abschließen, bevor Sie beginnen können.

Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern. Um über TLS eine Verbindung zu Amazon Keyspaces herzustellen, müssen Sie ein digitales Amazon-Zertifikat herunterladen und den Python-Treiber für die Verwendung von TLS konfigurieren.

Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -O
```

Note

Sie können das digitale Amazon-Zertifikat auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen, und Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -O
```

Stellen Sie mithilfe des Python-Treibers für Apache Cassandra und dienstspezifische Anmeldeinformationen eine Connect zu Amazon Keyspaces her

Das folgende Codebeispiel zeigt Ihnen, wie Sie mit einem Python-Client-Treiber und dienstspezifischen Anmeldeinformationen eine Verbindung zu Amazon Keyspaces herstellen.

```
from cassandra.cluster import Cluster
from ssl import SSLContext, PROTOCOL_TLSv1_2 , CERT_REQUIRED
from cassandra.auth import PlainTextAuthProvider

ssl_context = SSLContext(PROTOCOL_TLSv1_2 )
```

```
ssl_context.load_verify_locations('path_to_file/sf-class2-root.crt')
ssl_context.verify_mode = CERT_REQUIRED
auth_provider = PlainTextAuthProvider(username='ServiceUserName',
password='ServicePassword')
cluster = Cluster(['cassandra.us-east-2.amazonaws.com'], ssl_context=ssl_context,
auth_provider=auth_provider, port=9142)
session = cluster.connect()
r = session.execute('select * from system_schema.keyspaces')
print(r.current_rows)
```

Hinweise zur Verwendung:

1. "`path_to_file/sf-class2-root.crt`" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.
2. Stellen Sie sicher, dass der `ServiceUserName` `ServicePassword` mit dem Benutzernamen und dem Kennwort übereinstimmt, die Sie bei der Generierung der dienstspezifischen Anmeldeinformationen erhalten haben, indem Sie die Schritte bis [Generieren Sie dienstspezifische Anmeldeinformationen](#) ausführen.
3. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called "Service-Endpunkte"](#)

Stellen Sie mithilfe des DataStax Python-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her

Der folgende Abschnitt zeigt, wie Sie das SigV4-Authentifizierungs-Plugin für den DataStax Open-Source-Python-Treiber für Apache Cassandra verwenden, um auf Amazon Keyspaces (für Apache Cassandra) zuzugreifen.

Falls Sie dies noch nicht getan haben, beginnen Sie mit der Erstellung von Anmeldeinformationen für Ihre IAM-Rolle. Folgen Sie dabei den Schritten unter [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#) In diesem Tutorial werden temporäre Anmeldeinformationen verwendet, für die eine IAM-Rolle erforderlich ist. Weitere Informationen zu temporären Anmeldeinformationen finden Sie unter [the section called "Temporäre Anmeldeinformationen verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen"](#).

Fügen Sie dann das Python-SigV4-Authentifizierungs-Plugin aus dem [GitHub Repository](#) zu Ihrer Umgebung hinzu.

```
pip install cassandra-sigv4
```

Das folgende Codebeispiel zeigt, wie Sie mithilfe des DataStax Open-Source-Python-Treibers für Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Verbindung zu Amazon Keyspaces herstellen. Das Plugin hängt vom AWS SDK for Python (Boto3) ab. Es dient `boto3.session` zum Abrufen temporärer Anmeldeinformationen.

```

from cassandra.cluster import Cluster
from ssl import SSLContext, PROTOCOL_TLSv1_2 , CERT_REQUIRED
from cassandra.auth import PlainTextAuthProvider
import boto3
from cassandra_sigv4.auth import SigV4AuthProvider

ssl_context = SSLContext(PROTOCOL_TLSv1_2)
ssl_context.load_verify_locations('path_to_file/sf-class2-root.crt')
ssl_context.verify_mode = CERT_REQUIRED

# use this if you want to use Boto to set the session parameters.
boto_session = boto3.Session(aws_access_key_id="AKIAIOSFODNN7EXAMPLE",
                             aws_secret_access_key="wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
                             aws_session_token="AQoDYXdzEJr...<remainder of token>",
                             region_name="us-east-2")
auth_provider = SigV4AuthProvider(boto_session)

# Use this instead of the above line if you want to use the Default Credentials and not
  bother with a session.
# auth_provider = SigV4AuthProvider()

cluster = Cluster(['cassandra.us-east-2.amazonaws.com'], ssl_context=ssl_context,
                  auth_provider=auth_provider,
                  port=9142)
session = cluster.connect()
r = session.execute('select * from system_schema.keyspaces')
print(r.current_rows)

```

Hinweise zur Verwendung:

1. "*path_to_file/sf-class2-root.crt*" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.
2. *Stellen Sie sicher, dass aws_access_key_id, aws_secret_access_key und aws_session_token mit, und übereinstimmen, die Sie mit erhalten haben.*

Access Key Secret Access Key Session Token boto3.session Weitere Informationen finden Sie unter [Anmeldeinformationen](#) im AWS SDK for Python (Boto3).

3. Eine Liste der verfügbaren Endpunkte finden [the section called "Service-Endpunkte"](#) Sie unter.

Verwenden eines Cassandra Node.js Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

In diesem Abschnitt erfahren Sie, wie Sie mithilfe eines Node.js Client-Treibers eine Verbindung zu Amazon Keyspaces herstellen. Um Benutzern und Anwendungen Anmeldeinformationen für den programmatischen Zugriff auf Amazon Keyspaces-Ressourcen zur Verfügung zu stellen, können Sie einen der folgenden Schritte ausführen:

- Erstellen Sie dienstspezifische Anmeldeinformationen, die einem bestimmten AWS Identity and Access Management (IAM-) Benutzer zugeordnet sind.
- Aus Sicherheitsgründen empfehlen wir, IAM-Zugriffsschlüssel für IAM-Benutzer oder -Rollen zu erstellen, die für alle Dienste verwendet werden. AWS Das Amazon Keyspaces SigV4-Authentifizierungs-Plugin für Cassandra-Client-Treiber ermöglicht es Ihnen, Anrufe an Amazon Keyspaces mithilfe von IAM-Zugriffsschlüsseln anstelle von Benutzername und Passwort zu authentifizieren. Weitere Informationen finden Sie unter [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#).

Themen

- [Bevor Sie beginnen](#)
- [Stellen Sie mithilfe des DataStax Treibers Node.js für Apache Cassandra und dienstspezifische Anmeldeinformationen eine Connect zu Amazon Keyspaces her](#)
- [Stellen Sie mithilfe des Treibers DataStax Node.js für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her](#)

Bevor Sie beginnen

Sie müssen die folgende Aufgabe abschließen, bevor Sie beginnen können.

Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern. Um über TLS eine Verbindung zu Amazon Keyspaces herzustellen, müssen Sie ein digitales Amazon-Zertifikat herunterladen und den Python-Treiber für die Verwendung von TLS konfigurieren.

Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -0
```

Note

Sie können das digitale Amazon-Zertifikat auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen, und Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -0
```

Stellen Sie mithilfe des DataStax Treibers Node.js für Apache Cassandra und dienstspezifische Anmeldeinformationen eine Connect zu Amazon Keyspaces her

Konfigurieren Sie Ihren Treiber so, dass er das digitale Starfield-Zertifikat für TLS verwendet und sich mit dienstspezifischen Anmeldeinformationen authentifiziert. Beispielsweise:

```
const cassandra = require('cassandra-driver');
const fs = require('fs');
const auth = new cassandra.auth.PlainTextAuthProvider('ServiceUserName',
  'ServicePassword');
const sslOptions1 = {
  ca: [
    fs.readFileSync('path_to_file/sf-class2-root.crt', 'utf-8')],
  host: 'cassandra.us-west-2.amazonaws.com',
  rejectUnauthorized: true
};
const client = new cassandra.Client({
  contactPoints: ['cassandra.us-west-2.amazonaws.com'],
  localDataCenter: 'us-west-2',
  authProvider: auth,
  sslOptions: sslOptions1,
  protocolOptions: { port: 9142 }
});
const query = 'SELECT * FROM system_schema.keyspaces';
```

```
client.execute(query)
    .then( result => console.log('Row from Keyspaces %s',
    result.rows[0]))
    .catch( e=> console.log(`${e}`));
```

Hinweise zur Verwendung:

1. "*path_to_file*/sf-class2-root.crt" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.
2. Stellen Sie sicher, dass der *ServiceUserName* *ServicePassword* mit dem Benutzernamen und dem Kennwort übereinstimmt, die Sie bei der Generierung der dienstspezifischen Anmeldeinformationen erhalten haben, indem Sie die Schritte bis [Generieren Sie dienstspezifische Anmeldeinformationen](#) ausführen.
3. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called "Service-Endpunkte"](#)

Stellen Sie mithilfe des Treibers DataStax Node.js für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her

Der folgende Abschnitt zeigt, wie Sie das SigV4-Authentifizierungs-Plugin für den Open-Source-Treiber DataStax Node.js für Apache Cassandra verwenden, um auf Amazon Keyspaces (für Apache Cassandra) zuzugreifen.

Falls Sie dies noch nicht getan haben, erstellen Sie Anmeldeinformationen für Ihren IAM-Benutzer oder Ihre IAM-Rolle, indem Sie die Schritte unter befolgen. [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS "](#)

[Fügen Sie Ihrer Anwendung das SigV4-Authentifizierungs-Plug-In Node.js aus dem GitHub Repository hinzu.](#) Das Plugin unterstützt Version 4.x des DataStax Node.js -Treibers für Cassandra und ist vom AWS SDK für Node.js abhängig. Es dient `AWSCredentialsProvider` zum Abrufen von Anmeldeinformationen.

```
$ npm install aws-sigv4-auth-cassandra-plugin --save
```

Dieses Codebeispiel zeigt, wie eine regionsspezifische Instanz von `SigV4AuthProvider` als Authentifizierungsanbieter eingerichtet wird.

```
const cassandra = require('cassandra-driver');
const fs = require('fs');
```



```
const sigV4 = require('aws-sigv4-auth-cassandra-plugin');

const auth = new sigV4.SigV4AuthProvider({
  region: 'us-west-2',
  accessKeyId: 'AKIAIOSFODNN7EXAMPLE',
  secretAccessKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'});

const sslOptions1 = {
  ca: [
    fs.readFileSync('path_to_filecassandra/sf-class2-root.crt', 'utf-8')],
  host: 'cassandra.us-west-2.amazonaws.com',
  rejectUnauthorized: true
};

const client = new cassandra.Client({
  contactPoints: ['cassandra.us-west-2.amazonaws.com'],
  localDataCenter: 'us-west-2',
  authProvider: auth,
  sslOptions: sslOptions1,
  protocolOptions: { port: 9142 }
});

const query = 'SELECT * FROM system_schema.keyspaces';

client.execute(query).then(
  result => console.log('Row from Keyspaces %s', result.rows[0]))
  .catch( e=> console.log(`${e}`));
```

Hinweise zur Verwendung:

1. "*path_to_file*/sf-class2-root.crt" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.
2. Stellen Sie sicher, dass der *Zugriff KeyId* und das *Geheimnis AccessKey* mit dem Zugriffsschlüssel und dem geheimen Zugriffsschlüssel übereinstimmen, mit denen Sie ihn erhalten haben `AWSCredentialsProvider`. Weitere Informationen finden Sie unter [Setting Credentials in Node.js](#) im AWS SDK oder JavaScript in Node.js.
3. Informationen zum Speichern von Zugriffsschlüsseln außerhalb des Codes finden Sie unter Best Practices [the section called "Wie verwaltet man Zugangsschlüssel"](#).
4. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called "Service-Endpunkte"](#).

Verwenden eines Cassandra.NET-Core-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

In diesem Abschnitt erfahren Sie, wie Sie mithilfe eines .NET Core-Client-Treibers eine Verbindung zu Amazon Keyspaces herstellen. Die Einrichtungsschritte variieren je nach Umgebung und Betriebssystem. Möglicherweise müssen Sie sie entsprechend ändern. Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern. Um über TLS eine Verbindung zu Amazon Keyspaces herzustellen, müssen Sie ein digitales Starfield-Zertifikat herunterladen und Ihren Treiber für die Verwendung von TLS konfigurieren.

1. Laden Sie das Starfield-Zertifikat herunter und speichern Sie es in einem lokalen Verzeichnis, wobei Sie sich den Pfad notieren. Es folgt ein Beispiel mit PowerShell.

```
$client = new-object System.Net.WebClient
$client.DownloadFile("https://certs.secureserver.net/repository/sf-class2-
root.crt", "path_to_file\sf-class2-root.crt")
```

2. Installieren Sie den CassandRAC SharpDriver über Nuget mit der Nuget-Konsole.

```
PM> Install-Package CassandraCSharpDriver
```

3. Im folgenden Beispiel wird ein .NET Core C#-Konsolenprojekt verwendet, um eine Verbindung zu Amazon Keyspaces herzustellen und eine Abfrage auszuführen.

```
using Cassandra;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Security;
using System.Runtime.ConstrainedExecution;
using System.Security.Cryptography.X509Certificates;
using System.Text;
using System.Threading.Tasks;

namespace CSharpKeyspacesExample
{
    class Program
    {
        public Program(){}

        static void Main(string[] args)
```

```

    {
        X509Certificate2Collection certCollection = new
X509Certificate2Collection();
        X509Certificate2 amazoncert = new X509Certificate2(@"path_to_file\sf-
class2-root.crt");
        var userName = "ServiceUserName";
        var pwd = "ServicePassword";
        certCollection.Add(amazoncert);

        var awsEndpoint = "cassandra.us-east-2.amazonaws.com" ;

        var cluster = Cluster.Builder()
            .AddContactPoints(awsEndpoint)
            .WithPort(9142)
            .WithAuthProvider(new PlainTextAuthProvider(userName, pwd))
            .WithSSL(new
SSLOptions().SetCertificateCollection(certCollection))
            .Build();

        var session = cluster.Connect();
        var rs = session.Execute("SELECT * FROM system_schema.tables;");
        foreach (var row in rs)
        {
            var name = row.GetValue<String>("keyspace_name");
            Console.WriteLine(name);
        }
    }
}
}

```

Hinweise zur Verwendung:

- a. "*path_to_file*/sf-class2-root.crt" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.
- b. Stellen Sie sicher, dass der *ServiceUserName* *ServicePassword* mit dem Benutzernamen und dem Kennwort übereinstimmt, die Sie bei der Generierung der dienstspezifischen Anmeldeinformationen erhalten haben, indem Sie die Schritte bis [Generieren Sie dienstspezifische Anmeldeinformationen](#) ausführen.
- c. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called "Service-Endpunkte"](#)

Verwenden eines Cassandra Go-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

In diesem Abschnitt erfahren Sie, wie Sie mithilfe eines Go-Client-Treibers eine Verbindung zu Amazon Keyspaces herstellen. Um Benutzern und Anwendungen Anmeldeinformationen für den programmatischen Zugriff auf Amazon Keyspaces-Ressourcen zur Verfügung zu stellen, können Sie einen der folgenden Schritte ausführen:

- Erstellen Sie dienstspezifische Anmeldeinformationen, die einem bestimmten AWS Identity and Access Management (IAM-) Benutzer zugeordnet sind.
- Aus Sicherheitsgründen empfehlen wir, IAM-Zugriffsschlüssel für IAM-Benutzer und -Rollen zu erstellen, die für alle Dienste verwendet werden. AWS Das Amazon Keyspaces SigV4-Authentifizierungs-Plugin für Cassandra-Client-Treiber ermöglicht es Ihnen, Anrufe an Amazon Keyspaces mithilfe von IAM-Zugriffsschlüsseln anstelle von Benutzername und Passwort zu authentifizieren. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Themen

- [Bevor Sie beginnen](#)
- [Stellen Sie mithilfe des Gocql-Treibers für Apache Cassandra und dienstspezifische Anmeldeinformationen eine Connect zu Amazon Keyspaces her](#)
- [Stellen Sie mithilfe des Go-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plugins eine Connect zu Amazon Keyspaces her](#)

Bevor Sie beginnen

Sie müssen die folgende Aufgabe abschließen, bevor Sie beginnen können.

Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern. Um über TLS eine Verbindung zu Amazon Keyspaces herzustellen, müssen Sie ein digitales Amazon-Zertifikat herunterladen und den Python-Treiber für die Verwendung von TLS konfigurieren.

Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -0
```

Note

Sie können das digitale Amazon-Zertifikat auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen, und Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -0
```

Stellen Sie mithilfe des Gocql-Treibers für Apache Cassandra und dienstspezifische Anmeldeinformationen eine Connect zu Amazon Keyspaces her

1. Erstellen Sie ein Verzeichnis für die Anwendung.

```
mkdir ./gocqlexample
```

2. Navigieren Sie zum neuen Verzeichnis.

```
cd gocqlexample
```

3. Erstellen Sie eine Datei für Ihre Anwendung.

```
touch cqlapp.go
```

4. Laden Sie den Go-Treiber herunter.

```
go get github.com/gocql/gocql
```

5. Fügen Sie der Datei cqlapp.go den folgenden Beispielcode hinzu.

```
package main

import (
    "fmt"
    "github.com/gocql/gocql"
    "log"
)

func main() {
```

```
// add the Amazon Keyspaces service endpoint
cluster := gocql.NewCluster("cassandra.us-east-2.amazonaws.com")
cluster.Port=9142
// add your service specific credentials
cluster.Authenticator = gocql.PasswordAuthenticator{
    Username: "ServiceUserName",
    Password: "ServicePassword"}
// provide the path to the sf-class2-root.crt
cluster.SslOpts = &gocql.SslOptions{
    CaPath: "path_to_file/sf-class2-root.crt",
    EnableHostVerification: false,
}

// Override default Consistency to LocalQuorum
cluster.Consistency = gocql.LocalQuorum
cluster.DisableInitialHostLookup = false

session, err := cluster.CreateSession()
if err != nil {
    fmt.Println("err>", err)
}
defer session.Close()

// run a sample query from the system keyspace
var text string
iter := session.Query("SELECT keyspace_name FROM system_schema.tables;").Iter()
for iter.Scan(&text) {
    fmt.Println("keyspace_name:", text)
}
if err := iter.Close(); err != nil {
    log.Fatal(err)
}
session.Close()
}
```

Hinweise zur Verwendung:

- a. "*path_to_file*/sf-class2-root.crt" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.
- b. Stellen Sie sicher, dass der *ServiceUserName* *ServicePassword* mit dem Benutzernamen und dem Kennwort übereinstimmt, die Sie bei der Generierung der dienstspezifischen

Anmeldeinformationen erhalten haben, indem Sie die Schritte bis [Generieren Sie dienstspezifische Anmeldeinformationen](#) ausführen.

c. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called "Service-Endpunkte"](#)

6. Erstellen Sie das Programm.

```
go build cqlapp.go
```

7. Führen Sie das Programm aus.

```
./cqlapp
```

Stellen Sie mithilfe des Go-Treibers für Apache Cassandra und des SigV4-Authentifizierungs-Plug-ins eine Connect zu Amazon Keyspaces her

Das folgende Codebeispiel zeigt, wie Sie das SigV4-Authentifizierungs-Plugin für den Open-Source-Go-Treiber verwenden, um auf Amazon Keyspaces (für Apache Cassandra) zuzugreifen.

Falls Sie dies noch nicht getan haben, erstellen Sie Anmeldeinformationen für Ihren IAM-Benutzer oder Ihre IAM-Rolle, indem Sie die Schritte unter befolgen. [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#)

[Fügen Sie das Go SigV4-Authentifizierungs-Plugin aus dem Repository zu Ihrer Anwendung hinzu. GitHub](#) Das Plugin unterstützt Version 1.2.x des Open-Source-Go-Treibers für Cassandra und hängt vom SDK for Go ab. AWS

```
$ go mod init
$ go get github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin
```

In diesem Codebeispiel wird der Amazon Keyspaces-Endpunkt durch die `Cluster` Klasse dargestellt. Es verwendet die Eigenschaft `AwsAuthenticator` for the authenticator des Clusters, um Anmeldeinformationen abzurufen.

```
package main

import (
    "fmt"
    "github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin/sigv4"
    "github.com/gocql/gocql"
)
```

```

    "log"
)

func main() {
    // configuring the cluster options
    cluster := gocql.NewCluster("cassandra.us-west-2.amazonaws.com")
    cluster.Port=9142
    var auth sigv4.AwsAuthenticator = sigv4.NewAwsAuthenticator()
    auth.Region = "us-west-2"
    auth.AccessKeyId = "AKIAIOSFODNN7EXAMPLE"
    auth.SecretAccessKey = "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"

    cluster.Authenticator = auth

    cluster.SslOpts = &gocql.SslOptions{
        CaPath: "path_to_file/sf-class2-root.crt",
        EnableHostVerification: false,
    }
    cluster.Consistency = gocql.LocalQuorum
    cluster.DisableInitialHostLookup = false

    session, err := cluster.CreateSession()
    if err != nil {
        fmt.Println("err>", err)
        return
    }
    defer session.Close()

    // doing the query
    var text string
    iter := session.Query("SELECT keyspace_name FROM system_schema.tables;").Iter()
    for iter.Scan(&text) {
        fmt.Println("keyspace_name:", text)
    }
    if err := iter.Close(); err != nil {
        log.Fatal(err)
    }
}

```

Hinweise zur Verwendung:

1. "*path_to_file/sf-class2-root.crt*" Ersetzen Sie es durch den Pfad zu dem Zertifikat, das im ersten Schritt gespeichert wurde.

2. Stellen Sie sicher, dass die *AccessKeyID* und der *SecretAccessSchlüssel* mit dem Zugriffsschlüssel und dem geheimen Zugriffsschlüssel übereinstimmen, den Sie verwendet haben `AwsAuthenticator`. Weitere Informationen finden Sie unter [Konfiguration des AWS SDK for Go](#) in der AWS SDK for Go.
3. Informationen zum Speichern von Zugriffsschlüsseln außerhalb des Codes finden Sie unter Best Practices unter [the section called “Wie verwaltet man Zugangsschlüssel”](#).
4. Eine Liste der verfügbaren Endpunkte finden Sie unter [the section called “Service-Endpunkte”](#).

Verwenden eines Cassandra Perl-Client-Treibers für den programmgesteuerten Zugriff auf Amazon Keyspaces

In diesem Abschnitt erfahren Sie, wie Sie mithilfe eines Perl-Client-Treibers eine Verbindung zu Amazon Keyspaces herstellen. Für dieses Codebeispiel haben wir Perl 5 verwendet. Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern.

Important

Um eine sichere Verbindung herzustellen, verwenden unsere Codebeispiele das digitale Starfield-Zertifikat, um den Server zu authentifizieren, bevor die TLS-Verbindung hergestellt wird. Der Perl-Treiber validiert das Amazon SSL-Zertifikat des Servers nicht, was bedeutet, dass Sie nicht bestätigen können, dass Sie eine Verbindung zu Amazon Keyspaces herstellen. Der zweite Schritt, die Konfiguration des Treibers für die Verwendung von TLS bei der Verbindung zu Amazon Keyspaces, ist weiterhin erforderlich und stellt sicher, dass die zwischen dem Client und dem Server übertragenen Daten verschlüsselt werden.

1. Laden Sie den Cassandra DBI-Treiber von [herunter https://metacpan.org/pod/DBD::Cassandra](https://metacpan.org/pod/DBD::Cassandra) und installieren Sie ihn in Ihrer Perl-Umgebung. Die genauen Schritte hängen von der Umgebung ab. Das Folgende ist ein gängiges Beispiel.

```
cpanm DBD::Cassandra
```

2. Erstellen Sie eine Datei für Ihre Anwendung.

```
touch cqlapp.pl
```

3. Fügen Sie der Datei `cqlapp.pl` den folgenden Beispielcode hinzu.

```
use DBI;
my $user = "ServiceUserName";
my $password = "ServicePassword";
my $db = DBI->connect("dbi:Cassandra:host=cassandra.us-east-2.amazonaws.com;port=9142;tls=1;",
    $user, $password);

my $rows = $db->selectall_arrayref("select * from system_schema.keyspaces");
print "Found the following Keyspaces...\n";
for my $row (@$rows) {
    print join(" ",@$row['keyspace_name']),"\n";
}

$db->disconnect;
```

Important

Stellen Sie sicher, dass der *ServiceUserName* *ServicePassword* mit dem Benutzernamen und dem Passwort übereinstimmt, die Sie bei der Generierung der dienstspezifischen Anmeldeinformationen erhalten haben, indem Sie die Schritte bis ausführen. [Generieren Sie dienstspezifische Anmeldeinformationen](#)

Note

Eine Liste der verfügbaren Endpunkte finden Sie unter. [the section called "Service-Endpunkte"](#)

4. Führen Sie die Anwendung aus.

```
perl cqlapp.pl
```

Tutorial: Herstellen einer Verbindung mit Amazon Keyspaces vom Amazon Elastic Kubernetes Service

Dieses Tutorial führt Sie durch die Schritte, die zum Einrichten eines Amazon Elastic Kubernetes Service (Amazon EKS)-Clusters zum Hosten einer containerisierten Anwendung erforderlich sind, die mithilfe der SigV4-Authentifizierung eine Verbindung zu Amazon Keyspaces herstellt.

Amazon EKS ist ein verwalteter Service, der die Installation, den Betrieb und die Wartung Ihrer eigenen Kubernetes-Steuerebene überflüssig macht. [Kubernetes](#) ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von Anwendungen in Containern.

Das Tutorial enthält step-by-step Anleitungen zum Konfigurieren, Erstellen und Bereitstellen einer containerisierten Java-Anwendung in Amazon EKS. Im letzten Schritt führen Sie die Anwendung aus, um Daten in eine Amazon Keyspaces-Tabelle zu schreiben.

Themen

- [Tutorial-Voraussetzungen](#)
- [Schritt 1: Konfigurieren des Amazon-EKS-Clusters und Einrichten von IAM-Berechtigungen](#)
- [Schritt 2: Konfigurieren der Anwendung](#)
- [Schritt 3: Erstellen des Anwendungs-Images und Hochladen der Docker-Datei in Ihr Amazon-ECR-Repository](#)
- [Schritt 4: Bereitstellen der Anwendung in Amazon EKS und Schreiben von Daten in Ihre Amazon Keyspaces-Tabelle](#)
- [Schritt 5: \(Optional\) Bereinigen](#)

Tutorial-Voraussetzungen

Erstellen Sie die folgenden AWS Ressourcen, bevor Sie mit dem Tutorial beginnen können

1. Bevor Sie mit diesem Tutorial beginnen, folgen Sie den AWS Einrichtungsanweisungen unter [Zugreifen auf Amazon Keyspaces \(für Apache Cassandra\)](#). Zu diesen Schritten gehören die Registrierung für AWS und das Erstellen eines AWS Identity and Access Management (IAM)-Prinzips mit Zugriff auf Amazon Keyspaces.
2. Erstellen Sie einen Amazon Keyspaces Keyspace mit dem Namen `aws` und eine Tabelle mit dem Namen `user`, in den Sie von der containerisierten Anwendung schreiben können, die

später in Amazon EKS ausgeführt wird. Sie können dies entweder mit der AWS CLI oder mit der `tuncqlsh`.

AWS CLI

```
aws keyspaces create-keyspace --keyspace-name 'aws'
```

Um zu bestätigen, dass der Keyspace erstellt wurde, können Sie den folgenden Befehl verwenden.

```
aws keyspaces list-keyspaces
```

Um die Tabelle zu erstellen, können Sie den folgenden Befehl verwenden.

```
aws keyspaces create-table --keyspace-name 'aws' --table-name 'user' --schema-  
definition 'allColumns=[  
    {name=username,type=text}, {name=fname,type=text},  
{name=last_update_date,type=timestamp},{name=lname,type=text}],  
    partitionKeys=[{name=username}]'
```

Um zu bestätigen, dass Ihre Tabelle erstellt wurde, können Sie den folgenden Befehl verwenden.

```
aws keyspaces list-tables --keyspace-name 'aws'
```

Weitere Informationen finden Sie unter [Erstellen eines Keyspace](#) und [Erstellen einer Tabelle](#) in der AWS CLI -Befehlsreferenz.

cqlsh

```
CREATE KEYSPACE aws WITH replication = {'class': 'SimpleStrategy',  
    'replication_factor': '3'} AND durable_writes = true;  
CREATE TABLE aws.user (  
    username text PRIMARY KEY,  
    fname text,  
    last_update_date timestamp,  
    lname text  
);
```

Um zu überprüfen, ob Ihre Tabelle erstellt wurde, können Sie die folgende Anweisung verwenden.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = "aws";
```

Ihre Tabelle sollte in der Ausgabe dieser Anweisung aufgeführt sein. Beachten Sie, dass es bis zur Erstellung der Tabelle zu einer Verzögerung kommen kann. Weitere Informationen finden Sie unter [the section called "CREATE TABLE"](#).

- Erstellen Sie einen Amazon-EKS-Cluster mit einem Fargate-Linux-Knotentyp. Fargate ist eine Serverless-Compute-Engine, mit der Sie Kubernetes-Pods bereitstellen können, ohne Amazon-EC2 zu verwalten. Um diesem Tutorial zu folgen, ohne den Clusternamen in allen Beispielbefehlen aktualisieren zu müssen, erstellen Sie einen Cluster mit dem Namen `my-eks-cluster` gemäß den Anweisungen unter [Erste Schritte mit Amazon EKS im eksctl](#) Amazon-EKS-Benutzerhandbuch. Wenn Ihr Cluster erstellt wurde, stellen Sie sicher, dass Ihre Knoten und die beiden Standard-Pods ausgeführt werden und fehlerfrei sind. Sie können dies mit dem folgenden Befehl tun.

```
kubectl get pods -A -o wide
```

Sie sollten etwas sehen, das dieser Ausgabe ähnelt.

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE	IP
NODE				NOMINATED	NODE	READINESS
GATES						
kube-system	coredns-1234567890-abcde	1/1	Running	0	18m	
192.0.2.0	fargate-ip-192-0-2-0.region-code.compute.internal				<none>	
<none>						
kube-system	coredns-1234567890-12345	1/1	Running	0	18m	
192.0.2.1	fargate-ip-192-0-2-1.region-code.compute.internal				<none>	
<none>						

- Docker-Installation. Anweisungen zur Installation von Docker auf einer Amazon EC2-Instance finden Sie unter [Installieren von Docker](#) im Benutzerhandbuch für Amazon Elastic Container Registry.

Docker ist auf vielen verschiedenen Betriebssystemen verfügbar, darunter die meisten modernen Linux-Verteilungen wie Ubuntu und sogar macOS und Windows. Weitere Informationen

zur Installation von Docker unter einem bestimmten Betriebssystem finden Sie im [Docker-Installationshandbuch](#).

- Erstellen Sie ein Amazon-ECR-Repository. Amazon ECR ist ein AWS verwalteter Container-Image-Registry-Service, den Sie mit Ihrer bevorzugten CLI verwenden können, um Docker-Images zu pushen, abzurufen und zu verwalten. Weitere Informationen zu Amazon-ECR-Repositorys finden Sie im [Benutzerhandbuch für Amazon Elastic Container Registry](#). Sie können den folgenden Befehl verwenden, um ein Repository mit dem Namen zu erstellen `my-ecr-repository`.

```
aws ecr create-repository --repository-name my-ecr-repository
```

Fahren Sie nach Abschluss der erforderlichen Schritte mit fort [the section called “Schritt 1: Konfigurieren des Amazon-EKS-Clusters”](#).

Schritt 1: Konfigurieren des Amazon-EKS-Clusters und Einrichten von IAM-Berechtigungen

Konfigurieren Sie den Amazon-EKS-Cluster und erstellen Sie die IAM-Ressourcen, die erforderlich sind, damit ein Amazon-EKS-Servicekonto eine Verbindung zu Ihrer Amazon-Keyspaces-Tabelle herstellen kann

- Erstellen Sie einen Open ID Connect (OIDC)-Anbieter für den Amazon-EKS-Cluster. Dies ist erforderlich, um IAM-Rollen für Servicekonten zu verwenden. Weitere Informationen zu OIDC-Anbietern und deren Erstellung finden Sie unter [Erstellen eines IAM-OIDC-Anbieters für Ihren Cluster](#) im Amazon-EKS-Benutzerhandbuch.
 - Erstellen Sie einen IAM-OIDC-Identitätsanbieter für Ihren Cluster mit dem folgenden Befehl. In diesem Beispiel wird davon ausgegangen, dass Ihr Clustername lautet `my-eks-cluster`. Wenn Sie einen Cluster mit einem anderen Namen haben, denken Sie daran, den Namen in allen zukünftigen Befehlen zu aktualisieren.

```
eksctl utils associate-iam-oidc-provider --cluster my-eks-cluster --approve
```

- Bestätigen Sie mit dem folgenden Befehl, dass der OIDC-Identitätsanbieter bei IAM registriert wurde.

```
aws iam list-open-id-connect-providers --region aws-region
```

Die Ausgabe sollte in etwa so aussehen. Notieren Sie sich den Amazon-Ressourcennamen (ARN) des OIDC, den Sie im nächsten Schritt benötigen, wenn Sie eine Vertrauensrichtlinie für das Servicekonto erstellen.

```
{
  "OpenIDConnectProviderList": [
    ..
    {
      "Arn": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.aws-
region.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
    }
  ]
}
```

2. Erstellen Sie ein Servicekonto für den Amazon-EKS-Cluster. Servicekonten stellen eine Identität für Prozesse bereit, die in einem Pod ausgeführt werden. Ein Pod ist das kleinste und einfachste Kubernetes-Objekt, mit dem Sie eine containerisierte Anwendung bereitstellen können. Erstellen Sie als Nächstes eine IAM-Rolle, die das Servicekonto annehmen kann, um Berechtigungen für -Ressourcen zu erhalten. Sie können von einem Pod aus auf jeden AWS Service zugreifen, der für die Verwendung eines Servicekontos konfiguriert wurde, das eine IAM-Rolle mit Zugriffsberechtigungen für diesen Service übernehmen kann.
 - a. Erstellen Sie einen neuen Namespace für das Servicekonto. Ein Namespace hilft dabei, Cluster-Ressourcen zu isolieren, die für dieses Tutorial erstellt wurden. Sie können einen neuen Namespace mit dem folgenden Befehl erstellen.

```
kubectl create namespace my-eks-namespace
```

- b. Um einen benutzerdefinierten Namespace zu verwenden, müssen Sie ihn einem Fargate-Profil zuordnen. Der folgende Code ist ein Beispiel dafür.

```
eksctl create fargateprofile \
  --cluster my-eks-cluster \
  --name my-fargate-profile \
  --namespace my-eks-namespace \
  --labels *=*
```

- c. Erstellen Sie mit dem folgenden Befehl ein Servicekonto mit dem Namen `my-eks-serviceaccount` im Namespace `my-eks-namespace` für Ihren Amazon-EKS-Cluster.

```
cat >my-serviceaccount.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: my-eks-serviceaccount
  namespace: my-eks-namespace
EOF
kubectl apply -f my-serviceaccount.yaml
```

- d. Führen Sie den folgenden Befehl aus, um eine Vertrauensrichtliniendatei zu erstellen, die die IAM-Rolle anweist, Ihrem Servicekonto zu vertrauen. Diese Vertrauensstellung ist erforderlich, bevor ein Prinzipal eine Rolle übernehmen kann. Sie müssen die folgenden Änderungen an der Datei vornehmen:
- Geben Sie für den ARN ein `Principal`, den IAM an den `list-open-id-connect-providers` Befehl zurückgegeben hat. Der ARN enthält Ihre -Kontonummer und -Region.
 - Ersetzen Sie in der `-condition` Anweisung die AWS-Region und die OIDC-ID.
 - Vergewissern Sie sich, dass der Name und der Name des Servicekontos korrekt sind.

Sie müssen die Vertrauensrichtliniendatei im nächsten Schritt anfügen, wenn Sie die IAM-Rolle erstellen.

```
cat >trust-relationship.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.aws-region.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.aws-region.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:my-eks-
namespace:my-eks-serviceaccount",
```



```

        "oidc.eks.aws-region.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
    }
}
]
}
EOF

```

Optional: Sie können auch mehrere Einträge in den `StringLike` Bedingungen `StringEquals` oder hinzufügen, damit mehrere Servicekonten oder Namespaces die Rolle übernehmen können. Informationen dazu, wie Sie Ihrem Servicekonto erlauben können, eine IAM-Rolle in einem anderen AWS Konto anzunehmen, finden Sie unter [Kontoübergreifende IAM-Berechtigungen im Amazon-EKS-Benutzerhandbuch](#).

3. Erstellen Sie eine IAM-Rolle mit dem Namen `my-iam-role` für das Amazon-EKS-Servicekonto, das Sie übernehmen möchten. Hängen Sie die im letzten Schritt erstellte Vertrauensrichtliniendatei an die Rolle an. Die Vertrauensrichtlinie gibt das Servicekonto und den OIDC-Anbieter an, denen die IAM-Rolle vertrauen kann.

```
aws iam create-role --role-name my-iam-role --assume-role-policy-document file://trust-relationship.json --description "EKS service account role"
```

4. Weisen Sie Amazon Keyspaces die IAM-Rollenberechtigungen zu, indem Sie eine Zugriffsrichtlinie anfügen.
 - a. Fügen Sie eine Zugriffsrichtlinie an, um die Aktionen zu definieren, die die IAM-Rolle für bestimmte Amazon Keyspaces-Ressourcen ausführen kann. Für dieses Tutorial verwenden wir die - AWS verwaltete Richtlinie `AmazonKeyspacesFullAccess`, da unsere Anwendung Daten in Ihre Amazon Keyspaces-Tabelle schreiben wird. Als bewährte Methode wird jedoch empfohlen, benutzerdefinierte Zugriffsrichtlinien zu erstellen, die das Prinzip der geringsten Rechte implementieren. Weitere Informationen finden Sie unter [the section called "So funktioniert Amazon Keyspaces mit IAM"](#).

```
aws iam attach-role-policy --role-name my-iam-role --policy-arn=arn:aws:iam::aws:policy/AmazonKeyspacesFullAccess
```

Bestätigen Sie mit der folgenden Anweisung, dass die Richtlinie erfolgreich an die IAM-Rolle angefügt wurde.

```
aws iam list-attached-role-policies --role-name my-iam-role
```

Die Ausgabe sollte wie folgt aussehen.

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "AmazonKeyspacesFullAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AmazonKeyspacesFullAccess"
    }
  ]
}
```

- b. Kommentieren Sie das Servicekonto mit dem Amazon-Ressourcennamen (ARN) der IAM-Rolle, die es annehmen kann. Stellen Sie sicher, dass Sie den Rollen-ARN mit Ihrer Konto-ID aktualisieren.

```
kubectl annotate serviceaccount -n my-eks-namespace my-eks-serviceaccount eks.amazonaws.com/role-arn=arn:aws:iam::111122223333:role/my-iam-role
```

5. Vergewissern Sie sich, dass die IAM-Rolle und das Servicekonto korrekt konfiguriert sind.
 - a. Vergewissern Sie sich, dass die Vertrauensrichtlinie der IAM-Rolle mit der folgenden Anweisung korrekt konfiguriert ist.

```
aws iam get-role --role-name my-iam-role --query Role.AssumeRolePolicyDocument
```

Die Ausgabe sollte in etwa so aussehen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.aws-region.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
```

```

        "StringEquals": {
            "oidc.eks.aws-region/id/
EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
            "oidc.eks.aws-region.amazonaws.com/id/
EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:my-eks-
namespace:my-eks-serviceaccount"
        }
    }
}
]
}

```

- b. Vergewissern Sie sich, dass das Amazon-EKS-Servicekonto mit der IAM-Rolle versehen ist.

```
kubectl describe serviceaccount my-eks-serviceaccount -n my-eks-namespace
```

Die Ausgabe sollte in etwa so aussehen.

```

Name: my-eks-serviceaccount
Namespace:my-eks-namespace
Labels: <none>
Annotations: eks.amazonaws.com/role-arn: arn:aws:iam::111122223333:role/my-iam-
role
Image pull secrets: <none>
Mountable secrets: <none>
Tokens: <none>
[...]

```

Nachdem Sie das Amazon-EKS-Servicekonto und die IAM-Rolle erstellt und die erforderlichen Beziehungen und Berechtigungen konfiguriert haben, fahren Sie mit fort [the section called "Schritt 2: Konfigurieren der Anwendung"](#).

Schritt 2: Konfigurieren der Anwendung

In diesem Schritt erstellen Sie Ihre Anwendung, die mithilfe des SigV4-Plugins eine Verbindung zu Amazon Keyspaces herstellt. Sie können die Beispiel-Java-Anwendung im Amazon Keyspaces-Beispielcode-Repo auf [Github](#) anzeigen und herunterladen. Oder Sie können Ihre eigene Anwendung verwenden und sicherstellen, dass Sie alle Konfigurationsschritte ausführen.

Konfigurieren Sie Ihre Anwendung und fügen Sie die erforderlichen Abhängigkeiten hinzu.

1. Sie können die Java-Beispielanwendung herunterladen, indem Sie das Github-Repository mit dem folgenden Befehl klonen.

```
git clone https://github.com/aws-samples/amazon-keyspaces-examples.git
```

2. Entpacken Sie nach dem Herunterladen des Github-Repo die heruntergeladene Datei und navigieren Sie zum `resources` Verzeichnis zur `application.conf` Datei .
 - a. Anwendungskonfiguration

In diesem Schritt konfigurieren Sie das SigV4-Authentifizierungs-Plugin. Sie können das folgende Beispiel in Ihrer Anwendung verwenden. Falls noch nicht geschehen, müssen Sie Ihre IAM-Zugriffsschlüssel (Zugriffsschlüssel-ID und geheimer Zugriffsschlüssel) generieren und in Ihrer AWS Konfigurationsdatei oder als Umgebungsvariablen speichern. Detaillierte Anweisungen finden Sie unter [the section called “Erforderliche Anmeldeinformationen für die Authentifizierung AWS”](#). Aktualisieren Sie die AWS Region und den Service-Endpunkt für Amazon Keyspaces nach Bedarf. Weitere Service-Endpunkte finden Sie unter [the section called “Service-Endpunkte”](#). Ersetzen Sie den Truststore-Speicherort, den Truststore-Namen und das Truststore-Passwort durch Ihr eigenes.

```
datastax-java-driver {
  basic.contact-points = ["cassandra.aws-region.amazonaws.com:9142"]
  basic.load-balancing-policy.local-datacenter = "aws-region"
  advanced.auth-provider {
    class = software.aws.mcs.auth.SigV4AuthProvider
    aws-region = "aws-region"
  }
  advanced.ssl-engine-factory {
    class = DefaultSslEngineFactory
    truststore-path = "truststore_locationtruststore_name.jks"
    truststore-password = "truststore_password;"
  }
}
```

- b. Fügen Sie die Abhängigkeit des STS-Moduls hinzu.

Dies bietet die Möglichkeit, eine zu verwenden `WebIdentityTokenCredentialsProvider`, die die AWS Anmeldeinformationen zurückgibt, die die Anwendung bereitstellen muss, damit das

Servicekonto die IAM-Rolle übernehmen kann. Sie können dies auf der Grundlage des folgenden Beispiels tun.

```
<dependency>
  <groupId>com.amazonaws</groupId>
  <artifactId>aws-java-sdk-sts</artifactId>
  <version>1.11.717</version>
</dependency>
```

- c. Fügen Sie die SigV4-Abhängigkeit hinzu.

Dieses Paket implementiert das SigV4-Authentifizierungs-Plugin, das für die Authentifizierung bei Amazon Keyspaces benötigt wird

```
<dependency>
  <groupId>software.aws.mcs</groupId>
  <artifactId>aws-sigv4-auth-cassandra-java-driver-plugin</
artifactId>
  <version>4.0.3</version>
</dependency>
```

3. Fügen Sie eine Protokollierungsabhängigkeit hinzu.

Ohne Protokolle ist die Behebung von Verbindungsproblemen unmöglich. In diesem Tutorial verwenden wir `slf4j` als Protokollierungs-Framework und verwenden `logback.xml` um die Protokollausgabe zu speichern. Wir setzen die Protokollierungsebene auf `debug` um die Verbindung herzustellen. Sie können das folgende Beispiel verwenden, um die Abhängigkeit hinzuzufügen.

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>2.0.5</version>
</dependency>
```

Sie können den folgenden Codeausschnitt verwenden, um die Protokollierung zu konfigurieren.

```
<configuration>
  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">

    <encoder>
```

```
        <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</
pattern>
    </encoder>
</appender>

<root level="debug">
    <appender-ref ref="STDOUT" />
</root>
</configuration>
```

Note

Die `-debug` Ebene wird benötigt, um Verbindungsfehler zu untersuchen. Nachdem Sie von Ihrer Anwendung aus erfolgreich eine Verbindung zu Amazon Keyspaces hergestellt haben, können Sie die Protokollierungsstufe `warning` nach Bedarf in `info` oder ändern.

Schritt 3: Erstellen des Anwendungs-Images und Hochladen der Docker-Datei in Ihr Amazon-ECR-Repository

In diesem Schritt kompilieren Sie die Beispielanwendung, erstellen ein Docker-Image und übertragen das Image in Ihr Amazon-ECR-Repository.

Erstellen Sie Ihre Anwendung, erstellen Sie ein Docker-Image und senden Sie es an Amazon Elastic Container Registry

1. Legen Sie Umgebungsvariablen für den Build fest, die Ihre definieren AWS-Region. Ersetzen Sie die Regionen in den Beispielen durch Ihre eigenen.

```
export CASSANDRA_HOST=cassandra.aws-region.amazonaws.com:9142
export CASSANDRA_DC=aws-region
```

2. Kompilieren Sie Ihre Anwendung mit Apache Maven Version 3.6.3 oder höher mit dem folgenden Befehl.

```
mvn clean install
```

Dadurch wird eine JAR Datei mit allen im `target` Verzeichnis enthaltenen Abhängigkeiten erstellt.

3. Rufen Sie mit dem folgenden Befehl Ihren ECR-Repository-URI ab, der für den nächsten Schritt benötigt wird. Stellen Sie sicher, dass Sie die Region auf die Region aktualisieren, die Sie bereits verwendet haben.

```
aws ecr describe-repositories --region aws-region
```

Die Ausgabe sollte wie im folgenden Beispiel aussehen.

```
"repositories": [  
  {  
    "repositoryArn": "arn:aws:ecr:aws-region:111122223333:repository/my-ecr-  
repository",  
    "registryId": "111122223333",  
    "repositoryName": "my-ecr-repository",  
    "repositoryUri": "111122223333.dkr.ecr.aws-region.amazonaws.com/my-ecr-  
repository",  
    "createdAt": "2023-11-02T03:46:34+00:00",  
    "imageTagMutability": "MUTABLE",  
    "imageScanningConfiguration": {  
      "scanOnPush": false  
    },  
    "encryptionConfiguration": {  
      "encryptionType": "AES256"  
    }  
  },  
]
```

4. Erstellen Sie aus dem Stammverzeichnis der Anwendung das Docker-Image mit dem Repository-URI aus dem letzten Schritt. Ändern Sie die Docker-Datei nach Bedarf. Stellen Sie im Build-Befehl sicher, dass Sie Ihre Konto-ID ersetzen und AWS-Region auf die Region festlegen, in der sich das Amazon-ECR-Repository `my-ecr-repository` befindet.

```
docker build -t 111122223333.dkr.ecr.aws-region.amazonaws.com/my-ecr-  
repository:latest .
```

5. Rufen Sie ein Authentifizierungstoken ab, um das Docker-Image an Amazon ECR zu übertragen. Sie können dies mit dem folgenden Befehl tun.

```
aws ecr get-login-password --region aws-region | docker login --username AWS --  
password-stdin 111122223333.dkr.ecr.aws-region.amazonaws.com
```

6. Überprüfen Sie zunächst, ob in Ihrem Amazon-ECR-Repository Images vorhanden sind. Sie können folgenden Befehl verwenden.

```
aws ecr describe-images --repository-name my-ecr-repository --region aws-region
```

Verschieben Sie dann das Docker-Image in das Repo. Sie können folgenden Befehl verwenden.

```
docker push 111122223333.dkr.ecr.aws-region.amazonaws.com/my-ecr-repository:latest
```

Schritt 4: Bereitstellen der Anwendung in Amazon EKS und Schreiben von Daten in Ihre Amazon Keyspaces-Tabelle

In diesem Schritt des Tutorials konfigurieren Sie die Amazon-EKS-Bereitstellung für Ihre Anwendung und bestätigen, dass die Anwendung ausgeführt wird und eine Verbindung zu Amazon Keyspaces herstellen kann.

Um eine Anwendung in Amazon EKS bereitzustellen, müssen Sie alle relevanten Einstellungen in einer Datei namens `konfigurierendeployment.yaml`. Diese Datei wird dann von Amazon EKS verwendet, um die Anwendung bereitzustellen. Die Metadaten in der Datei sollten die folgenden Informationen enthalten:

- Anwendungsname der Name der Anwendung. Für dieses Tutorial verwenden wir `my-keyspaces-app`.
- Kubernetes-Namespace der Namespace des Amazon-EKS-Clusters. Für dieses Tutorial verwenden wir `my-eks-namespace`.
- Das Amazon-EKS-Servicekonto benennt den Namen des Amazon-EKS-Servicekontos. Für dieses Tutorial verwenden wir `my-eks-serviceaccount`.
- Image-Name der Name des Anwendungs-Images. Für dieses Tutorial verwenden wir `my-keyspaces-app`.
- Image-URI der Docker-Image-URI von Amazon ECR.
- -AWS Konto-ID Ihre - AWS Konto-ID.
- IAM-Rollen-ARN der ARN der IAM-Rolle, die für das zu übernehmende Servicekonto erstellt wurde. Für dieses Tutorial verwenden wir `my-iam-role`.
- AWS-Region des Amazon-EKS-Clusters, in dem AWS-Region Sie Ihren Amazon-EKS-Cluster erstellt haben.

In diesem Schritt stellen Sie die Anwendung bereit und führen sie aus, die eine Verbindung zu Amazon Keyspaces herstellt und Daten in die Tabelle schreibt.

1. Konfigurieren Sie die `deployment.yaml`-Datei. Sie müssen die folgenden Werte ersetzen:

- `name`
- `namespace`
- `serviceAccountName`
- `image`
- `AWS_ROLE_ARN` value
- Die AWS-Region in `CASSANDRA_HOST`
- `AWS_REGION`

Sie können die folgende Datei als Beispiel verwenden.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-keyspaces-app
  namespace: my-eks-namespace
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-keyspaces-app
  template:
    metadata:
      labels:
        app: my-keyspaces-app
    spec:
      serviceAccountName: my-eks-serviceaccount
      containers:
        - name: my-keyspaces-app
          image: 111122223333.dkr.ecr.aws-region.amazonaws.com/my-ecr-repository:latest
          ports:
            - containerPort: 8080
          env:
            - name: CASSANDRA_HOST
              value: "cassandra.aws-region.amazonaws.com:9142"
```

```
- name: CASSANDRA_DC
  value: "aws-region"
- name: AWS_WEB_IDENTITY_TOKEN_FILE
  value: /var/run/secrets/eks.amazonaws.com/serviceaccount/token
- name: AWS_ROLE_ARN
  value: "arn:aws:iam::111122223333:role/my-iam-role"
- name: AWS_REGION
  value: "aws-region"
```

2. Stellen Sie `deployment.yaml` bereit.

```
kubectl apply -f deployment.yaml
```

Die Ausgabe sollte wie folgt aussehen.

```
deployment.apps/my-keyspaces-app created
```

3. Überprüfen Sie den Status des Pods in Ihrem Namespace des Amazon-EKS-Clusters.

```
kubectl get pods -n my-eks-namespace
```

Die Ausgabe sollte in etwa wie in diesem Beispiel aussehen.

```
NAME                                READY STATUS RESTARTS AGE
my-keyspaces-app-123abcde4f-g5hij  1/1 Running 0 75s
```

Weitere Informationen finden Sie mit dem folgenden Befehl.

```
kubectl describe pod my-keyspaces-app-123abcde4f-g5hij -n my-eks-namespace
```

```
Name:                                my-keyspaces-app-123abcde4f-g5hij
Namespace:                            my-eks-namespace
Priority:                              2000001000
Priority Class Name:                   system-node-critical
Service Account:                       my-eks-serviceaccount
Node:                                  fargate-ip-192-168-102-209.ec2.internal/192.168.102.209
Start Time:                            Thu, 23 Nov 2023 12:15:43 +0000
Labels:                                app=my-keyspaces-app
                                         eks.amazonaws.com/fargate-profile=my-fargate-profile
```

```

pod-template-hash=6c56fccc56
Annotations:      CapacityProvisioned: 0.25vCPU 0.5GB
                  Logging: LoggingDisabled: LOGGING_CONFIGMAP_NOT_FOUND
Status:           Running
IP:               192.168.102.209
IPs:
  IP:             192.168.102.209
Controlled By:    ReplicaSet/my-keyspaces-app-6c56fccc56
Containers:
  my-keyspaces-app:
    Container ID:
      containerd://41ff7811d33ae4bc398755800abcdc132335d51d74f218ba81da0700a6f8c67b
    Image:         111122223333.dkr.ecr.aws-region.amazonaws.com/
  my_eks_repository:latest
    Image ID:      111122223333.dkr.ecr.aws-region.amazonaws.com/
  my_eks_repository@sha256:fd3c6430fc5251661efce99741c72c1b4b03061474940200d0524b84a951439c
    Port:          8080/TCP
    Host Port:     0/TCP
    State:         Running
      Started:     Thu, 23 Nov 2023 12:15:19 +0000
      Finished:    Thu, 23 Nov 2023 12:16:17 +0000
    Ready:         True
    Restart Count: 1
    Environment:
      CASSANDRA_HOST:      cassandra.aws-region.amazonaws.com:9142
      CASSANDRA_DC:        aws-region
      AWS_WEB_IDENTITY_TOKEN_FILE: /var/run/secrets/eks.amazonaws.com/
  serviceaccount/token
    AWS_ROLE_ARN:         arn:aws:iam::111122223333:role/my-iam-role
    AWS_REGION:           aws-region
    AWS_STS_REGIONAL_ENDPOINTS: regional
  Mounts:
    /var/run/secrets/eks.amazonaws.com/serviceaccount from aws-iam-token (ro)
    /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fssbf (ro)
Conditions:
  Type           Status
  Initialized     True
  Ready           True
  ContainersReady True
  PodScheduled    True
Volumes:
  aws-iam-token:
    Type:          Projected (a volume that contains injected data from
multiple sources)

```

```

    TokenExpirationSeconds: 86400
  kube-api-access-fssbf:
    Type: Projected (a volume that contains injected data from
multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
  QoS Class: BestEffort
  Node-Selectors: <none>
  Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for
300s
               node.kubernetes.io/unreachable:NoExecute op=Exists for
300s
  Events:
  Type      Reason            Age           From          Message
  ----      -
Warning    LoggingDisabled   2m13s        fargate-scheduler Disabled logging
because aws-logging configmap was not found. configmap "aws-logging" not found
Normal     Scheduled         89s          fargate-scheduler Successfully
assigned my-eks-namespace/my-keyspaces-app-6c56fccc56-mgs2m to fargate-
ip-192-168-102-209.ec2.internal
Normal     Pulled            75s          kubelet
Successfully pulled image "111122223333.dkr.ecr.aws-region.amazonaws.com/
my_eks_repository:latest" in 13.027s (13.027s including waiting)
Normal     Pulling           54s (x2 over 88s) kubelet        Pulling image
"111122223333.dkr.ecr.aws-region.amazonaws.com/my_eks_repository:latest"
Normal     Created           54s (x2 over 75s) kubelet        Created container
my-keyspaces-app
Normal     Pulled            54s          kubelet
Successfully pulled image "111122223333.dkr.ecr.aws-region.amazonaws.com/
my_eks_repository:latest" in 222ms (222ms including waiting)
Normal     Started           53s (x2 over 75s) kubelet        Started container
my-keyspaces-app

```

- Überprüfen Sie die Protokolle des Pods, um zu bestätigen, dass Ihre Anwendung ausgeführt wird und eine Verbindung zu Ihrer Amazon Keyspaces-Tabelle herstellen kann. Sie können dies mit dem folgenden Befehl tun. Stellen Sie sicher, dass Sie den Namen Ihrer Bereitstellung ersetzen.

```
kubectl logs -f my-keyspaces-app-123abcde4f-g5hij -n my-eks-namespace
```

Sie sollten Anwendungsprotokolleinträge sehen können, die die Verbindung zu Amazon Keyspaces bestätigen, wie im folgenden Beispiel.

```

2:47:20.553 [s0-admin-0] DEBUG c.d.o.d.i.c.metadata.MetadataManager
- [s0] Adding initial contact points [Node(endPoint=cassandra.aws-
region.amazonaws.com/1.222.333.44:9142, hostId=null, hashCode=e750d92)]
22:47:20.562 [s0-admin-1] DEBUG c.d.o.d.i.c.c.ControlConnection - [s0] Initializing
with event types [SCHEMA_CHANGE, STATUS_CHANGE, TOPOLOGY_CHANGE]
22:47:20.564 [s0-admin-1] DEBUG c.d.o.d.i.core.context.EventBus - [s0] Registering
com.datastax.oss.driver.internal.core.metadata.LoadBalancingPolicyWrapper$$Lambda
$812/0x00000000801105e88@769afb95 for class
com.datastax.oss.driver.internal.core.metadata.NodeStateEvent
22:47:20.566 [s0-admin-1] DEBUG c.d.o.d.i.c.c.ControlConnection -
[s0] Trying to establish a connection to Node(endPoint=cassandra.us-
east-1.amazonaws.com/1.222.333.44:9142, hostId=null, hashCode=e750d92)

```

- Führen Sie die folgende CQL-Abfrage für Ihre Amazon Keyspaces-Tabelle aus, um zu bestätigen, dass eine Datenzeile in Ihre Tabelle geschrieben wurde:

```
SELECT * from aws.user;
```

Die Ausgabe sollte folgendermaßen aussehen:

```

fname      | lname | username | last_update_date
-----+-----+-----+-----
random     | k     | test     | 2023-12-07 13:58:31.57+0000

```

Schritt 5: (Optional) Bereinigen

Gehen Sie wie folgt vor, um alle in diesem Tutorial erstellten Ressourcen zu entfernen.

Entfernen Sie die in diesem Tutorial erstellten Ressourcen

- Löschen Sie Ihre Bereitstellung. Sie können dazu den folgenden Befehl verwenden.

```
kubectl delete deployment my-keyspaces-app -n my-eks-namespace
```

- Löschen Sie den Amazon-EKS-Cluster und alle darin enthaltenen Pods. Dadurch werden auch zugehörige Ressourcen wie das Servicekonto und der OIDC-Identitätsanbieter gelöscht. Sie können dazu den folgenden Befehl verwenden.

```
eksctl delete cluster --name my-eks-cluster --region aws-region
```

3. Löschen Sie die für das Amazon-EKS-Servicekonto verwendete IAM-Rolle mit Zugriffsberechtigungen für Amazon Keyspaces. Zuerst müssen Sie die verwaltete Richtlinie entfernen, die der Rolle zugeordnet ist.

```
aws iam detach-role-policy --role-name my-iam-role --policy-arn
arn:aws:iam::aws:policy/AmazonKeyspacesFullAccess
```

Anschließend können Sie die Rolle mit dem folgenden Befehl löschen.

```
aws iam delete-role --role-name my-iam-role
```

Weitere Informationen finden Sie unter [Löschen einer IAM-Rolle \(AWS CLI\)](#) im IAM-Benutzerhandbuch.

4. Löschen Sie das Amazon-ECR-Repository einschließlich aller darin gespeicherten Images. Sie können dies mit dem folgenden Befehl tun.

```
aws ecr delete-repository \
  --repository-name my-ecr-repository \
  --force \
  --region aws-region
```

Beachten Sie, dass das `force` Flag erforderlich ist, um ein Repository zu löschen, das Images enthält. Um Ihr Image zuerst zu löschen, können Sie dies mit dem folgenden Befehl tun.

```
aws ecr batch-delete-image \
  --repository-name my-ecr-repository \
  --image-ids imageTag=latest \
  --region aws-region
```

Weitere Informationen finden Sie unter [Löschen eines Images](#) im Amazon Elastic Container Registry-Benutzerhandbuch.

5. Löschen Sie den Amazon Keyspaces Keyspace und die Tabelle. Durch das Löschen des Keyspace werden automatisch alle Tabellen in diesem Keyspace gelöscht. Sie können dazu eine der folgenden Optionen verwenden.

AWS CLI

```
aws keyspaces delete-keyspace --keyspace-name 'aws'
```

Um zu bestätigen, dass der Keyspace gelöscht wurde, können Sie den folgenden Befehl verwenden.

```
aws keyspaces list-keyspaces
```

Um zuerst die Tabelle zu löschen, können Sie den folgenden Befehl verwenden.

```
aws keyspaces delete-table --keyspace-name 'aws' --table-name 'user'
```

Um zu bestätigen, dass Ihre Tabelle gelöscht wurde, können Sie den folgenden Befehl verwenden.

```
aws keyspaces list-tables --keyspace-name 'aws'
```

Weitere Informationen finden Sie unter [Löschen von Keyspace](#) und [Tabelle löschen](#) in der AWS CLI -Befehlsreferenz.

cqlsh

```
DROP KEYSPACE IF EXISTS "aws";
```

Um zu überprüfen, ob Ihre Keyspaces gelöscht wurden, können Sie die folgende Anweisung verwenden.

```
SELECT * FROM system_schema.keyspaces ;
```

Ihr Keyspace sollte nicht in der Ausgabe dieser Anweisung aufgeführt werden. Beachten Sie, dass es zu einer Verzögerung kommen kann, bis die Keyspaces gelöscht werden. Weitere Informationen finden Sie unter [the section called "SCHLÜSSELRAUM LÖSCHEN"](#).

Um zuerst die Tabelle zu löschen, können Sie den folgenden Befehl verwenden.

```
DROP TABLE "aws.user"
```

Um zu bestätigen, dass Ihre Tabelle gelöscht wurde, können Sie den folgenden Befehl verwenden.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = "aws";
```

Ihre Tabelle sollte nicht in der Ausgabe dieser Anweisung aufgeführt sein. Beachten Sie, dass es zu einer Verzögerung kommen kann, bis die Tabelle gelöscht wird. Weitere Informationen finden Sie unter [the section called "DROP TABLE"](#).

Tutorial: Herstellen einer Verbindung zu Amazon Keyspaces mithilfe eines Schnittstellen-VPC-Endpunkts

Dieses Tutorial führt Sie durch die Einrichtung und Verwendung eines Schnittstellen-VPC-Endpunkts für Amazon Keyspaces.

Schnittstellen-VPC-Endpunkte ermöglichen die private Kommunikation zwischen Ihrer Virtual Private Cloud (VPC), die in Amazon VPC läuft, und Amazon Keyspaces. Schnittstelle, über die VPC-Endpunkte betrieben werden. Dabei handelt es sich um einen AWS Dienst AWS PrivateLink, der die private Kommunikation zwischen VPCs und Diensten ermöglicht. AWS Weitere Informationen finden Sie unter [the section called "Verwendung von -Schnittstellen-VPC-Endpunkten"](#).

Themen

- [Voraussetzungen und Überlegungen zum Tutorial](#)
- [Schritt 1: Starten einer Amazon-EC2-Instance](#)
- [Schritt 2: Konfigurieren Ihrer Amazon-EC2-Instance](#)
- [Schritt 3: Erstellen Sie einen VPC-Endpunkt für Amazon Keyspaces](#)
- [Schritt 4: Berechtigungen für die VPC-Endpunktverbindung konfigurieren](#)
- [Schritt 5: Konfigurieren Sie die Überwachung mit CloudWatch](#)
- [Schritt 6: \(Optional\) Bewährte Methoden zur Konfiguration der Größe des Verbindungspools für Ihre Anwendung](#)
- [Schritt 7: \(Optional\) Aufräumen](#)

Voraussetzungen und Überlegungen zum Tutorial

Bevor Sie mit diesem Tutorial beginnen, folgen Sie den Anweisungen zur AWS Einrichtung unter [Zugreifen auf Amazon Keyspaces \(für Apache Cassandra\)](#). Zu diesen Schritten gehören die Registrierung AWS und Erstellung eines AWS Identity and Access Management (IAM-) Principals mit Zugriff auf Amazon Keyspaces. Notieren Sie sich den Namen des IAM-Benutzers und die Zugriffsschlüssel, da Sie sie später in diesem Tutorial benötigen.

Erstellen Sie einen Schlüsselraum mit dem Namen myKeyspace und mindestens einer Tabelle, um die Verbindung mit dem VPC-Endpunkt weiter unten in diesem Tutorial zu testen. Eine ausführliche Anleitung finden Sie unter [Erste Schritte](#).

Nachdem Sie die erforderlichen Schritte abgeschlossen haben, fahren Sie mit fort [Schritt 1: Starten einer Amazon-EC2-Instance](#).

Schritt 1: Starten einer Amazon-EC2-Instance

In diesem Schritt starten Sie eine Amazon-EC2-Instance in Ihrer Standard-AWS-VPC. Anschließend können Sie einen VPC-Endpunkt für Amazon Keyspaces erstellen und verwenden.

So starten Sie eine Amazon EC2 Instance

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie Launch Instance aus und gehen Sie folgendermaßen vor:

Wählen Sie im Dashboard der EC2-Konsole im Feld Launch Instance die Option Launch Instance und dann aus den angezeigten Optionen Launch Instance aus.

Geben Sie unter Name und Tags für Name einen aussagekräftigen Namen für Ihre Instance ein.

Unter Anwendungs- und Betriebssystem-Images (Amazon Machine Image):

- Wählen Sie Schnellstart und dann Ubuntu. Dies ist das Betriebssystem für Ihre Instance.
- Unter Amazon Machine Image (AMI) können Sie das Standard-Image verwenden, für das das kostenlose Kontingent gilt. Ein Amazon Machine Image (AMI) dient als grundlegende Konfigurationsvorlage für Ihre Instance.

Unter Instanztyp:

- Wählen Sie aus der Liste Instance-Typ den Instance-Typ t2.micro aus, der standardmäßig ausgewählt ist.

Wählen Sie unter Schlüsselpaar (Anmeldung) für Name des Schlüsselpaars eine der folgenden Optionen für dieses Tutorial aus:

- Wenn Sie nicht über ein Amazon-EC2-Schlüsselpaar verfügen, wählen Sie Neues Schlüsselpaar erstellen aus und befolgen Sie die Anweisungen. Sie werden aufgefordert, eine private Schlüsseldatei (.pem-Datei) herunterzuladen. Sie benötigen diese Datei später, wenn Sie sich bei Ihrer Amazon EC2 EC2-Instance anmelden. Notieren Sie sich also den Dateipfad.
- Wenn Sie bereits über ein bestehendes Amazon-EC2-Schlüsselpaar verfügen, navigieren Sie zu Schlüsselpaar wählen und wählen Sie Ihr Schlüsselpaar aus der Liste aus. Beachten Sie, dass Sie bereits über die private Schlüsseldatei (.pem-Datei) verfügen müssen, um sich bei Ihrer Amazon-EC2-Instance anmelden zu können.

Unter Netzwerkeinstellungen:

- Wählen Sie Bearbeiten aus.
- Wählen Sie Bestehende Sicherheitsgruppe auswählen aus.
- Wählen Sie in der Liste der Sicherheitsgruppen default aus. Dies ist die Standard-Sicherheitsgruppe für Ihre VPC.

Fahren Sie mit der Zusammenfassung fort.

- Sehen Sie sich im Bereich Zusammenfassung eine Zusammenfassung Ihrer Instanzkonfiguration an. Sobald Sie bereit sind, wählen Sie Instance starten aus.
3. Wählen Sie auf dem Abschlussbildschirm für die neue Amazon EC2 EC2-Instance die Kachel Connect to Instance aus. Der nächste Bildschirm zeigt die erforderlichen Informationen und die erforderlichen Schritte, um eine Verbindung zu Ihrer neuen Instance herzustellen. Notieren Sie sich die folgenden Informationen:
- Der Beispielbefehl zum Schutz der Schlüsseldatei
 - Die Verbindungszeichenfolge
 - Der öffentliche IPv4-DNS-Name

Nachdem Sie die Informationen auf dieser Seite zur Kenntnis genommen haben, können Sie mit dem nächsten Schritt in diesem Tutorial fortfahren ([Schritt 2: Konfigurieren Ihrer Amazon-EC2-Instance](#)).

Note

Es dauert einige Minuten, bis Ihre Amazon-EC2-Instance verfügbar ist. Überprüfen Sie, ob Instance-Zustand `running` wird und alle Status-Kontrollen bestanden hat, bevor Sie mit dem nächsten Schritt fortfahren.

Schritt 2: Konfigurieren Ihrer Amazon-EC2-Instance

Wenn Ihre Amazon EC2 EC2-Instance verfügbar ist, können Sie sich bei ihr anmelden und sie für die erste Verwendung vorbereiten.

Note

Bei den folgenden Schritten wird davon ausgegangen, dass Sie von einem Computer aus, auf dem Linux ausgeführt wird, eine Verbindung zu Ihrer Amazon EC2 EC2-Instance herstellen. Weitere Verbindungsmöglichkeiten finden Sie unter [Connect to your Linux Instance](#) im Amazon EC2 EC2-Benutzerhandbuch.

So konfigurieren Sie Ihre Amazon EC2 EC2-Instance

1. Sie müssen eingehenden SSH-Verkehr zu Ihrer Amazon EC2 EC2-Instance autorisieren. Erstellen Sie dazu eine neue EC2-Sicherheitsgruppe und weisen Sie die Sicherheitsgruppe dann Ihrer EC2-Instance zu.
 - a. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
 - b. Wählen Sie Sicherheitsgruppen erstellen aus. Führen Sie im Fenster Sicherheitsgruppen erstellen Folgendes aus:
 - Name der Sicherheitsgruppe — Geben Sie einen Namen für Ihre Sicherheitsgruppe ein.
Beispiel: `my-ssh-access`

- Beschreibung — Geben Sie eine kurze Beschreibung für die Sicherheitsgruppe ein.
- VPC — Wählen Sie Ihre Standard-VPC.
- Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus und gehen Sie wie folgt vor:
 - Typ — Wählen Sie SSH.
 - Quelle — Wähle meine IP.
 - Wählen Sie Regel hinzufügen aus.

Bestätigen Sie unten auf der Seite die Konfigurationseinstellungen und wählen Sie Create Security Group aus.

- c. Wählen Sie im Navigationsbereich Instances aus.
 - d. Wählen Sie die Amazon-EC2-Instance aus, die Sie in [Schritt 1: Starten einer Amazon-EC2-Instance](#) gestartet haben.
 - e. Wählen Sie Aktionen, Sicherheit und dann Sicherheitsgruppen ändern aus.
 - f. Wählen Sie unter Sicherheitsgruppen ändern die Sicherheitsgruppe aus, die Sie zuvor in diesem Verfahren erstellt haben (z. B. my-ssh-access). Die vorhandene default-Sicherheitsgruppe sollte ebenfalls ausgewählt werden. Bestätigen Sie die Konfigurationseinstellungen und wählen Sie Sicherheitsgruppen zuweisen aus.
2. Verwenden Sie den folgenden Befehl, um Ihre private Schlüsseldatei vor Zugriff zu schützen. Wenn Sie diesen Schritt überspringen, schlägt die Verbindung fehl.

```
chmod 400 path_to_file/my-keypair.pem
```

3. Verwenden Sie den Befehl ssh, wie im folgenden Beispiel gezeigt, um sich bei der Amazon-EC2-Instance anzumelden.

```
ssh -i path_to_file/my-keypair.pem ubuntu@public-dns-name
```

Sie müssen Ihre private Schlüsseldatei (.pem-Datei) und den öffentlichen DNS-Namen Ihrer Instance angeben. (Siehe [Schritt 1: Starten einer Amazon-EC2-Instance](#)).

Die Anmelde-ID lautet ubuntu. Es ist kein Passwort erforderlich.

Weitere Informationen zum Zulassen von Verbindungen zu Ihrer Amazon EC2 EC2-Instance und AWS CLI Anweisungen finden Sie unter [Autorisieren von eingehendem Datenverkehr für Ihre Linux-Instances](#) im Amazon EC2 EC2-Benutzerhandbuch.

4. Laden Sie die neueste Version von herunter und installieren Sie sie. AWS Command Line Interface

a. Installieren unzip.

```
sudo apt install unzip
```

b. Laden Sie die zip Datei mit dem herunter AWS CLI.

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"
```

c. Entpacken Sie die Datei.

```
unzip awscliv2.zip
```

d. Installiere das AWS CLI.

```
sudo ./aws/install
```

e. Bestätigen Sie die Version der AWS CLI Installation.

```
aws --version
```

Die Ausgabe sollte in etwa wie folgt aussehen:

```
aws-cli/2.9.19 Python/3.9.11 Linux/5.15.0-1028-aws exe/x86_64.ubuntu.22 prompt/  
off
```

5. Konfigurieren Sie Ihre AWS Anmeldeinformationen, wie im folgenden Beispiel gezeigt. Geben Sie Ihre AWS Zugriffsschlüssel-ID, Ihren geheimen Schlüssel und den Namen der Standardregion ein, wenn Sie dazu aufgefordert werden.

```
aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-east-1
Default output format [None]:
```

6. Sie müssen eine `cqlsh` Verbindung zu Amazon Keyspaces verwenden, um zu bestätigen, dass Ihr VPC-Endpunkt korrekt konfiguriert wurde. Wenn Sie Ihre lokale Umgebung oder den Amazon Keyspaces CQL-Editor in der verwenden AWS Management Console, erfolgt die Verbindung automatisch über den öffentlichen Endpunkt statt über Ihren VPC-Endpunkt. Um Ihre VPC-Endpunktverbindung in diesem Tutorial `cqlsh` zu testen, folgen Sie den Anweisungen zur Einrichtung unter [Verwendung cqlsh, um eine Verbindung zu Amazon Keyspaces herzustellen](#)

Sie sind jetzt bereit, einen VPC-Endpunkt für Amazon Keyspaces zu erstellen.

Schritt 3: Erstellen Sie einen VPC-Endpunkt für Amazon Keyspaces

In diesem Schritt erstellen Sie einen VPC-Endpunkt für Amazon Keyspaces mithilfe von AWS CLI. Um den VPC-Endpunkt mithilfe der VPC-Konsole zu erstellen, können Sie den Anweisungen zum [Erstellen eines VPC-Endpunkts](#) im Handbuch folgen. `aws privatelink` Geben Sie ein, wenn Sie nach dem Dienstnamen filtern. **Cassandra**

So erstellen Sie einen VPC-Endpunkt mit dem AWS CLI

1. Bevor Sie beginnen, stellen Sie sicher, dass Sie über den öffentlichen Endpunkt mit Amazon Keyspaces kommunizieren können.

```
aws keyspaces list-tables --keyspace-name 'myKeyspace'
```

Die Ausgabe zeigt eine Liste von Amazon Keyspaces-Tabellen, die im angegebenen Schlüsselraum enthalten sind. Wenn Sie keine Tabellen haben, ist die Liste leer.

```
{
  "tables": [
    {
      "keyspaceName": "myKeyspace",
      "tableName": "myTable1",
      "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
catalog/table/myTable1"
    },
    {
      "keyspaceName": "myKeyspace",
```

```

        "tableName": "myTable2",
        "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
catalog/table/myTable2"
    }
]
}

```

2. Stellen Sie sicher, dass Amazon Keyspaces ein verfügbarer Service für die Erstellung von VPC-Endpunkten in der aktuellen Region ist. AWS (Der Befehl wird in Fettschrift angezeigt, gefolgt von einer Beispielausgabe.)

```

aws ec2 describe-vpc-endpoint-services

{
  "ServiceNames": [
    "com.amazonaws.us-east-1.cassandra",
    "com.amazonaws.us-east-1.cassandra-fips"
  ]
}

```

In der Beispielausgabe ist Amazon Keyspaces einer der verfügbaren Services, sodass Sie mit der Erstellung eines VPC-Endpunkts dafür fortfahren können.

3. Ermitteln Sie den VPC-Bezeichner.

```

aws ec2 describe-vpcs

{
  "Vpcs": [
    {
      "VpcId": "vpc-a1234bcd",
      "InstanceTenancy": "default",
      "State": "available",
      "DhcpOptionsId": "dopt-8454b7e1",
      "CidrBlock": "111.31.0.0/16",
      "IsDefault": true
    }
  ]
}

```

In der Beispielausgabe ist die VPC-ID `vpc-a1234bcd`.

4. Verwenden Sie einen Filter, um Details zu den Subnetzen der VPC zu sammeln.

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-a1234bcd"

{
  {
    "Subnets":[
      {
        "AvailabilityZone":"us-east-1a",
        "AvailabilityZoneId":"use2-az1",
        "AvailableIpAddressCount":4085,
        "CidrBlock":"111.31.0.0/20",
        "DefaultForAz":true,
        "MapPublicIpOnLaunch":true,
        "MapCustomerOwnedIpOnLaunch":false,
        "State":"available",
        "SubnetId":"subnet-920aacf9",
        "VpcId":"vpc-a1234bcd",
        "OwnerId":"111122223333",
        "AssignIpv6AddressOnCreation":false,
        "Ipv6CidrBlockAssociationSet":[

        ],
        "SubnetArn":"arn:aws:ec2:us-east-1:111122223333:subnet/subnet-920aacf9",
        "EnableDns64":false,
        "Ipv6Native":false,
        "PrivateDnsNameOptionsOnLaunch":{"
          "HostnameType":"ip-name",
          "EnableResourceNameDnsARecord":false,
          "EnableResourceNameDnsAAAARecord":false
        }
      },
      {
        "AvailabilityZone":"us-east-1c",
        "AvailabilityZoneId":"use2-az3",
        "AvailableIpAddressCount":4085,
        "CidrBlock":"111.31.32.0/20",
        "DefaultForAz":true,
        "MapPublicIpOnLaunch":true,
        "MapCustomerOwnedIpOnLaunch":false,
        "State":"available",
        "SubnetId":"subnet-4c713600",
        "VpcId":"vpc-a1234bcd",
        "OwnerId":"111122223333",
        "AssignIpv6AddressOnCreation":false,
```



```

    "Ipv6CidrBlockAssociationSet":[
    ],
    "SubnetArn":"arn:aws:ec2:us-east-1:111122223333:subnet/subnet-4c713600",
    "EnableDns64":false,
    "Ipv6Native":false,
    "PrivateDnsNameOptionsOnLaunch":{"
      "HostnameType":"ip-name",
      "EnableResourceNameDnsARecord":false,
      "EnableResourceNameDnsAAAARecord":false
    }
  },
  {
    "AvailabilityZone":"us-east-1b",
    "AvailabilityZoneId":"use2-az2",
    "AvailableIpAddressCount":4086,
    "CidrBlock":"111.31.16.0/20",
    "DefaultForAz":true,
    "MapPublicIpOnLaunch":true,
  }
]
}

```

In der Beispielausgabe sind zwei Subnetz-IDs verfügbar: `und. subnet-920aacf9`
`subnet-4c713600`

- Erstellen des VPC-Endpunkts Geben Sie für den Parameter `--vpc-id` die VPC-ID aus dem vorherigen Schritt an. Geben Sie für den `--subnet-id` Parameter die Subnetz-IDs aus dem vorherigen Schritt an. Verwenden Sie den `--vpc-endpoint-type` Parameter, um den Endpunkt als Schnittstelle zu definieren. Weitere Informationen zu dem Befehl finden Sie [create-vpc-endpoint](#) in der AWS CLI Befehlsreferenz.

```

aws ec2 create-vpc-endpoint --vpc-endpoint-type Interface --vpc-id vpc-a1234bcd
--service-name com.amazonaws.us-east-1.cassandra --subnet-id subnet-920aacf9
subnet-4c713600

{
  "VpcEndpoint": {
    "VpcEndpointId": "vpce-000ab1cdef23456789",
    "VpcEndpointType": "Interface",
    "VpcId": "vpc-a1234bcd",
  }
}

```

```
"ServiceName": "com.amazonaws.us-east-1.cassandra",
"State": "pending",
"RouteTableIds": [],
"SubnetIds": [
  "subnet-920aacf9",
  "subnet-4c713600"
],
"Groups": [
  {
    "GroupId": "sg-ac1b0e8d",
    "GroupName": "default"
  }
],
"IpAddressType": "ipv4",
"DnsOptions": {
  "DnsRecordIpType": "ipv4"
},
"PrivateDnsEnabled": true,
"RequesterManaged": false,
"NetworkInterfaceIds": [
  "eni-043c30c78196ad82e",
  "eni-06ce37e3fd878d9fa"
],
"DnsEntries": [
  {
    "DnsName": "vpce-000ab1cdef23456789-m2b22rtz.cassandra.us-
east-1.vpce.amazonaws.com",
    "HostedZoneId": "Z7HUB22UULQXV"
  },
  {
    "DnsName": "vpce-000ab1cdef23456789-m2b22rtz-us-
east-1a.cassandra.us-east-1.vpce.amazonaws.com",
    "HostedZoneId": "Z7HUB22UULQXV"
  },
  {
    "DnsName": "vpce-000ab1cdef23456789-m2b22rtz-us-
east-1c.cassandra.us-east-1.vpce.amazonaws.com",
    "HostedZoneId": "Z7HUB22UULQXV"
  },
  {
    "DnsName": "vpce-000ab1cdef23456789-m2b22rtz-us-
east-1b.cassandra.us-east-1.vpce.amazonaws.com",
    "HostedZoneId": "Z7HUB22UULQXV"
  }
],
```

```
    {
      "DnsName": "vpce-000ab1cdef23456789-m2b22rtz-us-
east-1d.cassandra.us-east-1.vpce.amazonaws.com",
      "HostedZoneId": "Z7HUB22UULQXV"
    },
    {
      "DnsName": "cassandra.us-east-1.amazonaws.com",
      "HostedZoneId": "ZONEIDPENDING"
    }
  ],
  "CreationTimestamp": "2023-01-27T16:12:36.834000+00:00",
  "OwnerId": "111122223333"
}
}
```

Schritt 4: Berechtigungen für die VPC-Endpunktverbindung konfigurieren

Die Verfahren in diesem Schritt zeigen, wie Regeln und Berechtigungen für die Verwendung des VPC-Endpunkts mit Amazon Keyspaces konfiguriert werden.

Um eine eingehende Regel für den neuen Endpunkt zu konfigurieren, um eingehenden TCP-Verkehr zuzulassen

1. Wählen Sie in der Amazon VPC-Konsole im linken Bereich Endpoints und dann den Endpoint aus, den Sie im vorherigen Schritt erstellt haben.
2. Wählen Sie Sicherheitsgruppen und dann die Sicherheitsgruppe aus, die diesem Endpunkt zugeordnet ist.
3. Wählen Sie Regeln für eingehenden Datenverkehr und anschließend Regeln für eingehenden Datenverkehr bearbeiten aus.
4. Fügen Sie eine Regel für eingehenden Datenverkehr mit dem Typ CQLSH/CASSANDRA hinzu. Dadurch wird der Portbereich automatisch auf 9142 festgelegt.
5. Um die neue Regel für eingehenden Datenverkehr zu speichern, wählen Sie Regeln speichern.

Um IAM-Benutzerberechtigungen zu konfigurieren

1. Vergewissern Sie sich, dass der IAM-Benutzer, der für die Verbindung zu Amazon Keyspaces verwendet wurde, über die entsprechenden Berechtigungen verfügt. In AWS Identity and Access Management (IAM) können Sie die AWS verwaltete Richtlinie verwenden, `AmazonKeyspacesReadOnlyAccess` um dem IAM-Benutzer Lesezugriff auf Amazon Keyspaces zu gewähren.
 - a. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter `https://console.aws.amazon.com/iam/`.](https://console.aws.amazon.com/iam/)
 - b. Wählen Sie auf dem IAM-Konsolen-Dashboard Users (Benutzer) und anschließend Ihren IAM-Benutzer aus der Liste aus.
 - c. Wählen Sie auf der Seite Summary (Übersicht) die Option Add permissions (Berechtigungen hinzufügen) aus.
 - d. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen).
 - e. Wählen Sie in der Liste der Richtlinien die Option `AmazonKeyspacesReadOnlyAccess` und anschließend Next: Review aus.
 - f. Wählen Sie Add permissions (Berechtigungen hinzufügen) aus.
2. Stellen Sie sicher, dass Sie über den VPC-Endpunkt auf Amazon Keyspaces zugreifen können.

```
aws keyspaces list-tables --keyspace-name 'my_keyspace'
```

Wenn Sie möchten, können Sie einige andere AWS CLI Befehle für Amazon Keyspaces ausprobieren. Weitere Informationen finden Sie in der [AWS CLI -Befehlsreferenz](#).

Note

Die Mindestberechtigungen, die ein IAM-Benutzer oder eine IAM-Rolle für den Zugriff auf Amazon Keyspaces benötigt, sind Leseberechtigungen für die Systemtabelle, wie in der folgenden Richtlinie dargestellt. Weitere Informationen zu richtlinienbasierten Berechtigungen finden Sie unter [the section called "Beispiele für identitätsbasierte Richtlinien"](#)

```
{  
  "Version": "2012-10-17",
```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "cassandra:Select"
        ],
        "Resource": [
          "arn:aws:cassandra:us-east-1:555555555555:/keyspace/system*"
        ]
      }
    ]
  }
}

```

3. Gewähren Sie dem IAM-Benutzer Lesezugriff auf die Amazon EC2 EC2-Instance mit der VPC.

Wenn Sie Amazon Keyspaces mit VPC-Endpunkten verwenden, müssen Sie dem IAM-Benutzer oder der IAM-Rolle, der auf Amazon Keyspaces zugreift, nur Leseberechtigungen für Ihre Amazon EC2 EC2-Instance und die VPC gewähren, um Endpunkt- und Netzwerkschnittstellendaten zu sammeln. Amazon Keyspaces speichert diese Informationen in der `system.peers` Tabelle und verwendet sie zur Verwaltung von Verbindungen.

Note

Die verwalteten Richtlinien `AmazonKeyspacesReadOnlyAccess_v2` `AmazonKeyspacesFullAccess` enthalten die erforderlichen Berechtigungen, damit Amazon Keyspaces auf die Amazon EC2 EC2-Instance zugreifen kann, um Informationen über verfügbare Schnittstellen-VPC-Endpunkte zu lesen.

- a. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.](https://console.aws.amazon.com/iam/)
- b. Wählen Sie im Dashboard der IAM-Konsole die Option Richtlinien aus.
- c. Wählen Sie Richtlinie erstellen und dann die Registerkarte JSON aus.
- d. Kopieren Sie die folgende Richtlinie und wählen Sie Weiter: Tags.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListVPCEndpoints",

```

```

    "Effect": "Allow",
    "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcEndpoints"
    ],
    "Resource": "*"
  }
]
}

```

- e. Wählen Sie Weiter: Überprüfen, geben Sie den Namen `keyspacesVPCendpoint` für die Richtlinie ein und wählen Sie Richtlinie erstellen aus.
 - f. Wählen Sie auf dem IAM-Konsolen-Dashboard Users (Benutzer) und anschließend Ihren IAM-Benutzer aus der Liste aus.
 - g. Wählen Sie auf der Seite Summary (Übersicht) die Option Add permissions (Berechtigungen hinzufügen) aus.
 - h. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen).
 - i. Wählen Sie in der Liste der Richtlinien `KeySpaceSvpcEndpoint` und dann Weiter: Überprüfen aus.
 - j. Wählen Sie Add permissions (Berechtigungen hinzufügen) aus.
4. Um zu überprüfen, ob die Amazon `system.peers` Keyspaces-Tabelle mit VPC-Informationen aktualisiert wird, führen Sie die folgende Abfrage von Ihrer Amazon EC2 EC2-Instance aus mit `cqlsh`. Wenn Sie in Schritt 2 noch nicht `cqlsh` auf Ihrer Amazon EC2 EC2-Instance installiert haben, folgen Sie den Anweisungen unter [the section called "Verwenden des cqlsh-expansion"](#).

```
SELECT peer FROM system.peers;
```

Die Ausgabe gibt Knoten mit privaten IP-Adressen zurück, abhängig von Ihrer VPC- und Subnetzkonfiguration in Ihrer AWS Region.

```
peer
-----
112.11.22.123
112.11.22.124
112.11.22.125
```

Note

Sie müssen eine `cqlsh` Verbindung zu Amazon Keyspaces verwenden, um zu bestätigen, dass Ihr VPC-Endpoint korrekt konfiguriert wurde. Wenn Sie Ihre lokale Umgebung oder den Amazon Keyspaces CQL-Editor in der verwenden AWS Management Console, erfolgt die Verbindung automatisch über den öffentlichen Endpunkt statt über Ihren VPC-Endpoint. Wenn Sie neun IP-Adressen sehen, sind dies die Einträge, die Amazon Keyspaces automatisch in die `system.peers` Tabelle für öffentliche Endpunktverbindungen schreibt.

Schritt 5: Konfigurieren Sie die Überwachung mit CloudWatch

In diesem Schritt erfahren Sie, wie Sie Amazon verwenden CloudWatch , um die VPC-Endpointverbindung zu Amazon Keyspaces zu überwachen.

AWS PrivateLink veröffentlicht Datenpunkte zu CloudWatch etwa Ihren Schnittstellen-Endpunkten. Mit den Metriken können Sie überprüfen, ob Ihr System die erwartete Leistung zeigt. Der `AWS/PrivateLinkEndpoints` Namespace in CloudWatch enthält die Metriken für Schnittstellenendpunkte. Weitere Informationen finden Sie im Handbuch unter [CloudWatch AWS PrivateLink Metriken für](#).AWS PrivateLink

So erstellen Sie ein CloudWatch Dashboard mit VPC-Endpointmetriken

1. Öffnen Sie die CloudWatch Konsole unter. <https://console.aws.amazon.com/cloudwatch/>
2. Wählen Sie im Navigationsbereich Dashboards aus. Wählen Sie dann Dashboard erstellen. Geben Sie einen Namen für das Dashboard ein und wählen Sie Erstellen.
3. Wählen Sie unter Widget hinzufügen die Option Zahl aus.
4. Wählen Sie unter Metrics die Option AWS/ PrivateLink Endpoints aus.
5. Wählen Sie Endpunkttyp, Dienstname, VPC-Endpoint-ID, VPC-ID aus.
6. Wählen Sie die Metriken `ActiveConnections` und `NewConnections` dann Widget erstellen aus.
7. Speichern Sie das Dashboard.

Die `ActiveConnections` Metrik ist definiert als die Anzahl der gleichzeitigen aktiven Verbindungen, die der Endpunkt in der letzten Minute empfangen hat. Die `NewConnections` Metrik ist definiert als die Anzahl der neuen Verbindungen, die in der letzten Minute über den Endpunkt hergestellt wurden.

Weitere Informationen zum Erstellen von Dashboards finden Sie unter [Dashboard erstellen](#) im CloudWatch Benutzerhandbuch.

Schritt 6: (Optional) Bewährte Methoden zur Konfiguration der Größe des Verbindungspools für Ihre Anwendung

In diesem Abschnitt wird beschrieben, wie Sie die ideale Größe des Verbindungspools anhand der Anforderungen an den Abfragedurchsatz Ihrer Anwendung ermitteln können.

Amazon Keyspaces erlaubt maximal 3.000 CQL-Abfragen pro Sekunde pro TCP-Verbindung. Die Anzahl der Verbindungen, die ein Treiber mit Amazon Keyspaces herstellen kann, ist also praktisch unbegrenzt. Wir empfehlen Ihnen jedoch, die Größe des Verbindungspools an die Anforderungen Ihrer Anwendung anzupassen und die verfügbaren Endpunkte zu berücksichtigen, wenn Sie Amazon Keyspaces mit VPC-Endpunktverbindungen verwenden.

Sie konfigurieren die Größe des Verbindungspools im Client-Treiber. Basierend auf einer lokalen Poolgröße von 2 und einem VPC-Schnittstellenendpunkt, der über 3 Availability Zones erstellt wurde, richtet der Treiber beispielsweise 6 Verbindungen für Abfragen ein (insgesamt 7, einschließlich einer Kontrollverbindung). Mit diesen 6 Verbindungen können Sie maximal 18.000 CQL-Abfragen pro Sekunde unterstützen.

Wenn Ihre Anwendung 40.000 CQL-Abfragen pro Sekunde unterstützen muss, gehen Sie von der Anzahl der Abfragen aus, die erforderlich sind, um die erforderliche Größe des Verbindungspools zu ermitteln. Um 40.000 CQL-Abfragen pro Sekunde zu unterstützen, müssen Sie die Größe des lokalen Pools auf mindestens 5 konfigurieren, was mindestens 45.000 CQL-Abfragen pro Sekunde unterstützt.

Mithilfe der `PerConnectionRequestRateExceeded` CloudWatch Metrik im Namespace können Sie überwachen, ob Sie das Kontingent für die maximale Anzahl von Vorgängen pro Sekunde und Verbindung überschreiten. `AWS/Cassandra` Die `PerConnectionRequestRateExceeded` Metrik zeigt die Anzahl der Anfragen an Amazon Keyspaces, die das Kontingent für die Anforderungsrate pro Verbindung überschreiten.

Die Codebeispiele in diesem Schritt zeigen, wie Sie das Verbindungspooling schätzen und konfigurieren, wenn Sie VPC-Endpunkte mit Schnittstellen verwenden.

Java

Sie können die Anzahl der Verbindungen pro Pool im Java-Treiber konfigurieren. Ein vollständiges Beispiel für eine Verbindung mit einem Java-Clienttreiber finden Sie unter [the section called “Verwenden eines Cassandra-Java-Client-Treibers”](#).

Wenn der Client-Treiber gestartet wird, wird zunächst die Steuerverbindung für Verwaltungsaufgaben hergestellt, z. B. für Schema- und Topologieänderungen. Dann werden die zusätzlichen Verbindungen erstellt.

Im folgenden Beispiel ist die Treiberkonfiguration für die Größe des lokalen Pools mit 2 angegeben. Wenn der VPC-Endpoint über 3 Subnetze innerhalb der VPC erstellt wird, ergibt dies 7 NewConnections in CloudWatch für den Schnittstellenendpunkt, wie in der folgenden Formel dargestellt.

```
NewConnections = 3 (VPC subnet endpoints created across) * 2 (pool size) + 1
( control connection)
```

```
datastax-java-driver {

    basic.contact-points = [ "cassandra.us-east-1.amazonaws.com:9142" ]
    advanced.auth-provider{
        class = PlainTextAuthProvider
        username = "ServiceUserName"
        password = "ServicePassword"
    }
    basic.load-balancing-policy {
        local-datacenter = "us-east-1"
        slow-replica-avoidance = false
    }

    advanced.ssl-engine-factory {
        class = DefaultSslEngineFactory
        truststore-path = "./src/main/resources/cassandra_truststore.jks"
        truststore-password = "my_password"
        hostname-validation = false
    }
    advanced.connection {
        pool.local.size = 2
    }
}
```

Wenn die Anzahl der aktiven Verbindungen nicht Ihrer konfigurierten Poolgröße (Aggregation über Subnetze) +1 Kontrollverbindung entspricht, verhindert etwas, dass die Verbindungen erstellt werden.

Node.js

Sie können die Anzahl der Verbindungen pro Pool im Treiber Node.js konfigurieren. Ein vollständiges Beispiel für eine Clienttreiberverbindung vom Typ Node.js finden Sie unter [the section called “Verwenden eines Cassandra Node.js Client-Treibers”](#).

Im folgenden Codebeispiel ist die Treiberkonfiguration für die Größe des lokalen Pools als 1 angegeben. Wenn der VPC-Endpoint über 4 Subnetze innerhalb der VPC erstellt wird, ergibt dies 5 `NewConnections` in CloudWatch für den Schnittstellenendpunkt, wie in der folgenden Formel dargestellt.

```
NewConnections = 4 (VPC subnet endpoints created across) * 1 (pool size) + 1
( control connection)
```

```
const cassandra = require('cassandra-driver');
const fs = require('fs');
const types = cassandra.types;
const auth = new cassandra.auth.PlainTextAuthProvider('ServiceUserName',
  'ServicePassword');
const sslOptions1 = {
  ca: [
    fs.readFileSync('/home/ec2-user/sf-class2-root.crt', 'utf-8')],
  host: 'cassandra.us-east-1.amazonaws.com',
  rejectUnauthorized: true
};
const client = new cassandra.Client({
  contactPoints: ['cassandra.us-east-1.amazonaws.com'],
  localDataCenter: 'us-east-1',
  pooling: { coreConnectionsPerHost: { [types.distance.local]:
1 } },
  consistency: types.consistencies.localQuorum,
  queryOptions: { isIdempotent: true },
  authProvider: auth,
  sslOptions: sslOptions1,
  protocolOptions: { port: 9142 }
});
```

Schritt 7: (Optional) Aufräumen

Gehen Sie wie folgt vor, wenn Sie die Ressourcen löschen möchten, die Sie in diesem Tutorial erstellt haben.

So entfernen Sie Ihren VPC-Endpunkt für Amazon Keyspaces

1. Melden Sie sich bei Ihrer Amazon-EC2-Instance an.
2. Ermitteln Sie die VPC-Endpunkt-ID, die für Amazon Keyspaces verwendet wird. Wenn Sie die `grep` Parameter weglassen, werden VPC-Endpunktinformationen für alle Dienste angezeigt.

```
aws ec2 describe-vpc-endpoint-services | grep ServiceName | grep cassandra

{
  "VpcEndpoint": {
    "PolicyDocument": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}\",
    "VpcId": "vpc-0bbc736e",
    "State": "available",
    "ServiceName": "com.amazonaws.us-east-1.cassandra",
    "RouteTableIds": [],
    "VpcEndpointId": "vpce-9b15e2f2",
    "CreationTimestamp": "2017-07-26T22:00:14Z"
  }
}
```

In der Beispielausgabe ist die ID des VPC-Endpunkts `vpce-9b15e2f2`.

3. Löschen Sie den VPC-Endpunkt.

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-9b15e2f2

{
  "Unsuccessful": []
}
```

Das leere Array `[]` zeigt den Erfolg an (keine fehlgeschlagenen Anforderungen).

So beenden Sie Ihre Amazon-EC2-Instance

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.

2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie Ihre Amazon EC2-Instance aus.
4. Wählen Sie Actions, Instance State und anschließend Terminate aus.
5. Wählen Sie im Bestätigungsfenster Yes, Terminate aus.

Konfigurieren von für den Keyspaces

Sie können separate Ressourcen erstellen und verwenden AWS-Konten, um Ressourcen zu isolieren und in verschiedenen Umgebungen, z. B. in der Entwicklung und Produktion, zu verwenden.

Dieses Thema führt Sie durch den kontoübergreifenden Zugriff für Amazon Keyspaces mithilfe von Schnittstellen-VPC-Endpunkten in einem Amazon Virtual Private Cloud. Weitere Informationen zum kontoübergreifenden IAM-Zugriff finden Sie [unter Beispielszenario](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu Amazon Keyspaces und privaten VPC-Endpunkten finden Sie unter [the section called "Verwendung von -Schnittstellen-VPC-Endpunkten"](#).

Themen

- [Konfigurieren von für den Keyspaces den VPC](#)
- [Konfigurieren von für den Keyspaces den VPC](#)

Konfigurieren von für den Keyspaces den VPC

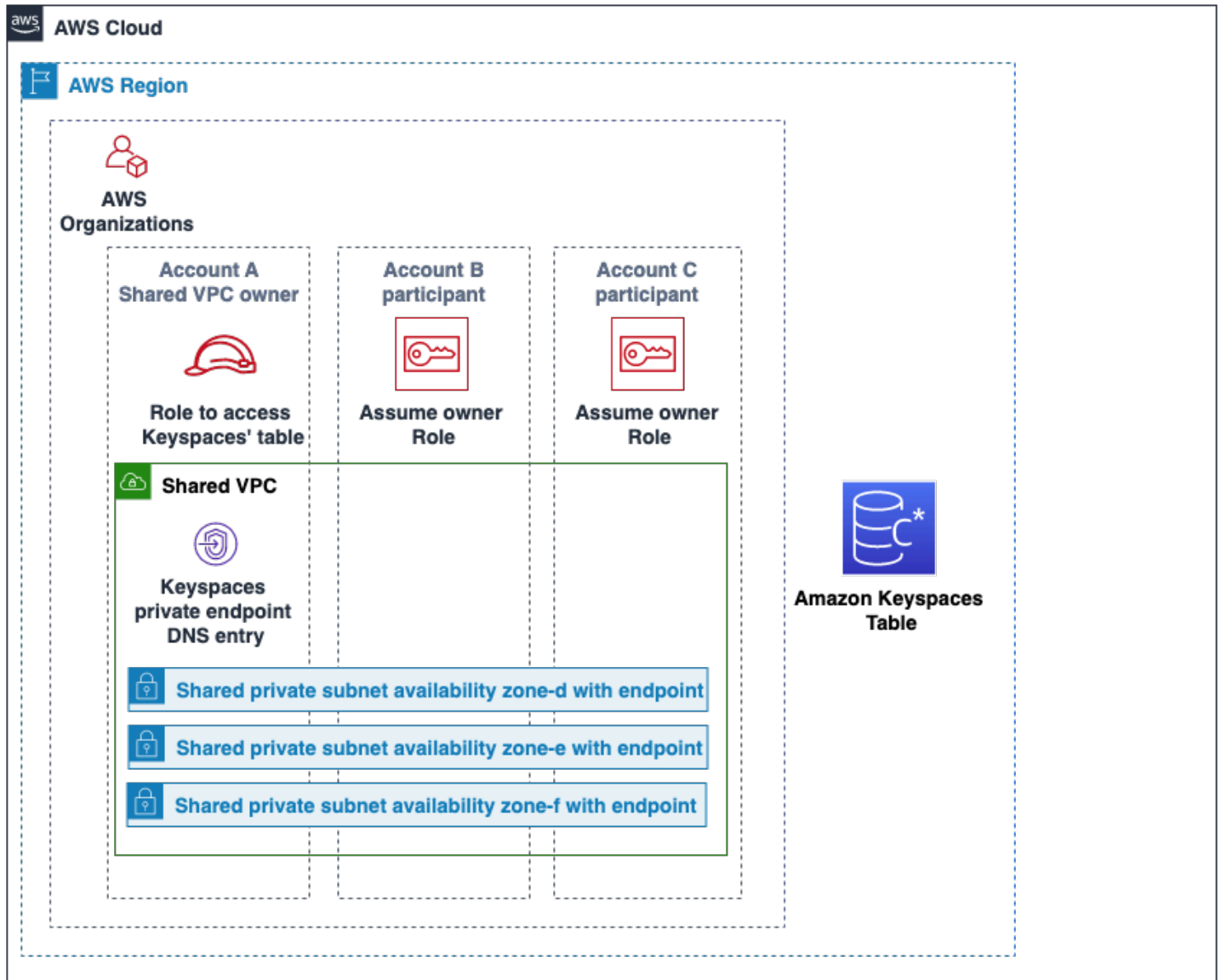
Sie können verschiedene erstellen AWS-Konten, um Ressourcen von Anwendungen zu trennen. Sie können beispielsweise ein Konto für Ihre Amazon Keyspaces-Tabellen, ein anderes Konto für Anwendungen in einer Entwicklungsumgebung und ein weiteres Konto für Anwendungen in einer Produktionsumgebung erstellen. Dieses Thema führt Sie durch die Konfigurationsschritte, die erforderlich sind, um den kontoübergreifenden Zugriff für Amazon Keyspaces mithilfe von Schnittstellen-VPC-Endpunkten in einer gemeinsam genutzten VPC einzurichten.

Eine ausführliche Anleitung zur Konfiguration eines VPC-Endpunkts für Amazon Keyspaces finden Sie unter [the section called "Schritt 3: Erstellen Sie einen VPC-Endpunkt für Amazon Keyspaces"](#).

In diesem Beispiel verwenden wir die folgenden drei Konten in einer gemeinsamen VPC:

- Account A— Dieses Konto enthält die Infrastruktur, einschließlich der VPC-Endpunkte, der VPC-Subnetze und Amazon Keyspaces-Tabellen.

- Account B— Dieses Konto enthält eine Anwendung in einer Entwicklungsumgebung, die eine Verbindung zur Amazon Keyspaces-Tabelle herstellen mussAccount A.
- Account C— Dieses Konto enthält eine Anwendung in einer Produktionsumgebung, die eine Verbindung zur Amazon Keyspaces-Tabelle herstellen mussAccount A.



Account A ist das Konto, das die Ressourcen enthält, auf die Account B und auf die zugegriffen werdenAccount C muss, ebensoAccount A das Treuhandkonto. Account B undAccount C sind die Konten bei den Auftraggebern, die Zugriff auf die Ressourcen benötigenAccount A, alsoAccount B undAccount C sind die vertrauenswürdigen Konten. Das vertrauenswürdige Konto

gewährt den vertrauenswürdigen Konten die Berechtigungen, indem es sich eine IAM-Rolle teilt. Im folgenden Verfahren werden die erforderlichen Konfigurationsschritte in Account A.

Konfiguration für Account A

1. Wird verwendet AWS Resource Access Manager, um eine Ressourcenfreigabe für das Subnetz zu erstellen und das private Subnetz mit Account B und zu teilen Account C.

Account B und Account C kann nun Ressourcen in dem Subnetz sehen und erstellen, das mit ihnen geteilt wurde.

2. Erstellen Sie einen privaten VPC-Endpunkt von Amazon Keyspaces, unterstützt von AWS PrivateLink. Dadurch werden mehrere Endpunkte in gemeinsam genutzten Subnetzen und DNS-Einträge für den Amazon Keyspaces-Serviceendpunkt erstellt.
3. Erstellen Sie einen Amazon Keyspaces-Schlüsselraum und eine Tabelle.
4. Erstellen Sie eine IAM-Rolle, die vollen Zugriff auf die Amazon Keyspaces-Tabelle und Lesezugriff auf die Amazon Keyspaces-Systemtabellen hat und die Amazon EC2-VPC-Ressourcen beschreiben kann, wie im folgenden Richtlinienbeispiel gezeigt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountAccess",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcEndpoints",
        "cassandra:*"
      ],
      "Resource": "*"
    }
  ]
}
```

5. Konfigurieren Sie die Vertrauensrichtlinie der IAM-Rolle, die Account B und als vertrauenswürdige Konten angenommen werden Account C können, wie im folgenden Beispiel gezeigt.

```
{
  "Version": "2012-10-17",
```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::111111111111:root"
        },
        "Action": "sts:AssumeRole",
        "Condition": {}
      }
    ]
  }

```

Weitere Informationen zu -Konten finden Sie unter [Kontoubergreifende Richtlinien](#) im IAM-Benutzerhandbuch.

Konfiguration in **Account B** und **Account C**

1. Erstellen Sie in **Account B** und **Account C** neue Rollen und fügen Sie die folgende Richtlinie hinzu, die es dem Principal ermöglicht, die in **Account A** erstellte gemeinsame Rolle zu übernehmen.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Dem Principal zu erlauben, die gemeinsame Rolle zu übernehmen, wird mithilfe der `AssumeRole` API von AWS Security Token Service (AWS STS) implementiert. Weitere Informationen finden Sie unter [Gewähren des des des des des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto](#) im IAM-Benutzerhandbuch.

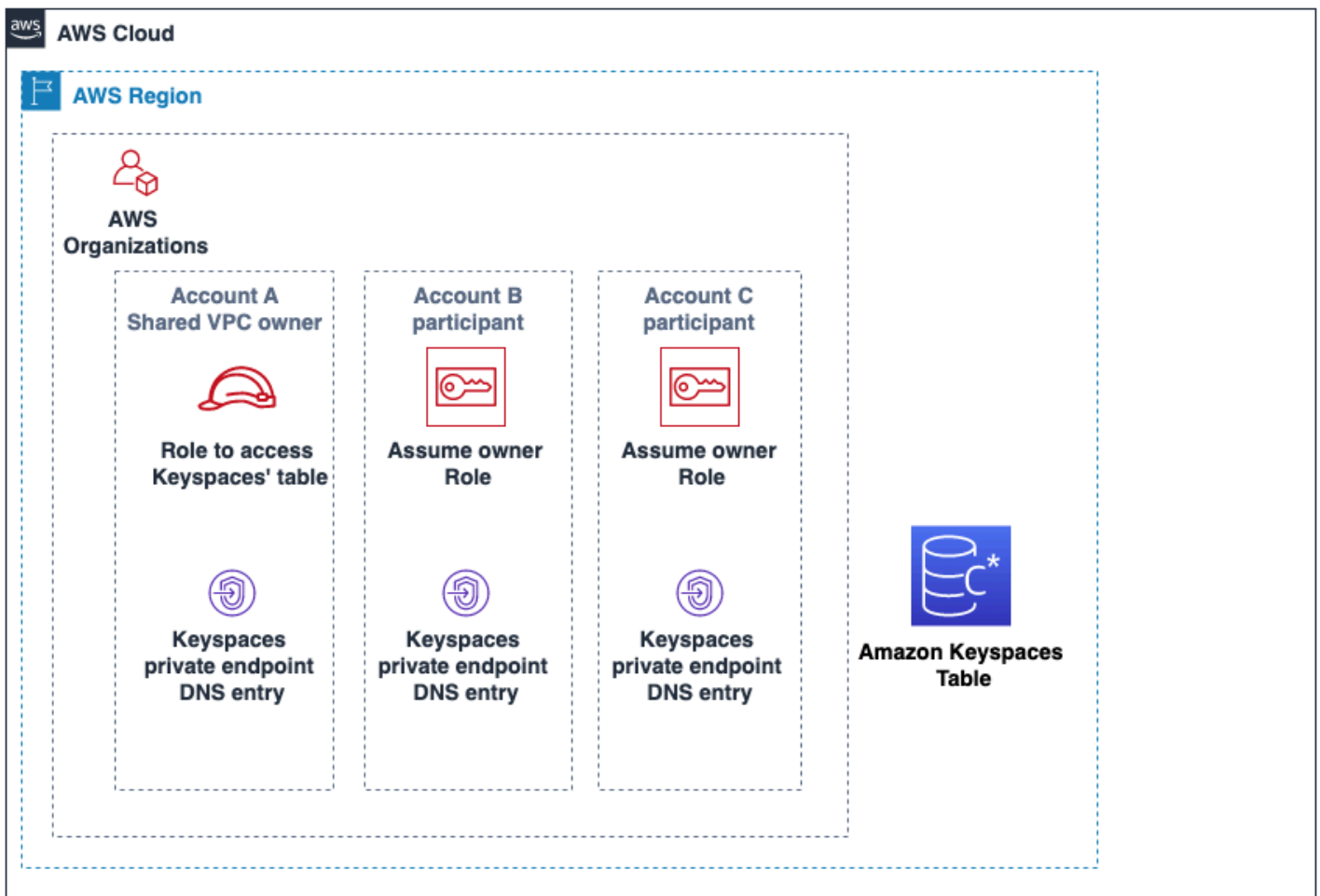
2. In **Account B** und **Account C** können Sie Anwendungen erstellen, die das SIGV4-Authentifizierungs-Plugin verwenden, wodurch eine Anwendung die gemeinsame Rolle

übernehmen kann, um Account A über den VPC-Endpunkt in der gemeinsam genutzten VPC eine Verbindung zur Amazon Keyspaces-Tabelle herzustellen. Weitere Informationen zum SIGV4-Authentifizierungs-Plug-in finden Sie unter [the section called “Anmeldeinformationen erstellen”](#).

Konfigurieren von für den Keyspaces den VPC

Wenn die Amazon Keyspaces-Tabelle und der private VPC-Endpunkt unterschiedlichen Konten gehören, sich aber keine VPC teilen, können Anwendungen mithilfe von VPC-Endpunkten trotzdem kontoübergreifende Verbindungen herstellen. Weil die Konten die VPC-Endpunkte, Account A, nicht gemeinsam nutzen und ihre eigenen VPC-Endpunkte Account C benötigen. Für den Cassandra-Client-Treiber erscheint Amazon Keyspaces wie ein einzelner Knoten statt wie ein Cluster mit mehreren Knoten. Nach der Verbindung erreicht der Client-Treiber den DNS-Server, der einen der verfügbaren Endpunkte in der VPC des Kontos zurückgibt.

Sie können auch über verschiedene Konten hinweg auf Amazon Keyspaces-Tabellen zugreifen, ohne einen gemeinsamen VPC-Endpunkt zu haben, indem Sie den öffentlichen Endpunkt verwenden oder in jedem Konto einen privaten VPC-Endpunkt bereitstellen. Wenn Sie keine gemeinsam genutzte VPC verwenden, benötigt jedes Konto seinen eigenen VPC-Endpunkt. In diesem Account A Beispiel Account C benötigen und ihre eigenen VPC-Endpunkte Account B, um auf die Tabelle zugreifen zu können Account A. Wenn Sie VPC-Endpunkte in dieser Konfiguration verwenden, erscheint Amazon Keyspaces für den Cassandra-Client-Treiber als Cluster mit einem einzelnen Knoten und nicht als Cluster mit mehreren Knoten. Nach der Verbindung erreicht der Client-Treiber den DNS-Server, der einen der verfügbaren Endpunkte in der VPC des Kontos zurückgibt. Der Client-Treiber kann jedoch nicht auf die `system.peers` Tabelle zugreifen, um zusätzliche Endpunkte zu ermitteln. Da weniger Hosts verfügbar sind, stellt der Treiber weniger Verbindungen her. Um dies anzupassen, erhöhen Sie die Verbindungspool-Einstellung des Treibers um den Faktor drei.



Erste Schritte mit Amazon Keyspaces (für Apache Cassandra)

Dieses Tutorial ist für Sie, wenn Sie Apache Cassandra und Amazon Keyspaces (für Apache Cassandra) noch nicht kennen. In diesem Tutorial installieren Sie alle Programme und Treiber, die Sie für die erfolgreiche Verwendung von Amazon Keyspaces benötigen.

Tutorials zur programmgesteuerten Verbindung mit Amazon Keyspaces mithilfe verschiedener Cassandra-Client-Treiber finden Sie unter. [the section called “Verwenden eines Cassandra-Client-Treibers”](#)

Themen

- [Voraussetzungen und Überlegungen zum Tutorial](#)
- [Tutorial Schritt 1: Einen Schlüsselraum und eine Tabelle in Amazon Keyspaces erstellen](#)
- [Tutorial Schritt 2: Daten erstellen, lesen, aktualisieren und löschen \(CRUD\)](#)
- [Tutorial Schritt 3: Löschen Sie eine Tabelle und einen Schlüsselraum in Amazon Keyspaces](#)

Voraussetzungen und Überlegungen zum Tutorial

Bevor Sie mit diesem Tutorial beginnen, folgen Sie den Anweisungen zur AWS Einrichtung unter. [Zugreifen auf Amazon Keyspaces \(für Apache Cassandra\)](#) Zu diesen Schritten gehören die Registrierung AWS und Erstellung eines AWS Identity and Access Management (IAM-) Benutzers mit Zugriff auf Amazon Keyspaces.

Wenn Sie das Tutorial mit `cqlsh` einem für Apache 2.0 lizenzierten Cassandra-Client-Treiber abschließen, befolgen Sie außerdem die Installationsanweisungen unter. [Verwendung `cqlsh`, um eine Verbindung zu Amazon Keyspaces herzustellen](#)

Nachdem Sie die erforderlichen Schritte abgeschlossen haben, fahren Sie mit fort [Tutorial Schritt 1: Einen Schlüsselraum und eine Tabelle in Amazon Keyspaces erstellen](#).

Tutorial Schritt 1: Einen Schlüsselraum und eine Tabelle in Amazon Keyspaces erstellen

In diesem Abschnitt erstellen Sie einen Schlüsselraum und fügen ihm mithilfe der Konsole eine Tabelle hinzu.

Note

Bevor Sie beginnen, stellen Sie sicher, dass Sie alle [Voraussetzungen für das Tutorial](#) konfiguriert haben.

Themen

- [Einen Schlüsselraum erstellen](#)
- [Erstellen einer Tabelle](#)

Einen Schlüsselraum erstellen

Ein Schlüsselraum gruppiert verwandte Tabellen, die für eine oder mehrere Anwendungen relevant sind. Ein Schlüsselraum enthält eine oder mehrere Tabellen und definiert die Replikationsstrategie für alle darin enthaltenen Tabellen. Weitere Informationen zu Schlüsselräumen finden Sie in den folgenden Themen:

- Mit Schlüsselräumen arbeiten: [the section called “Erstellen von Keyspaces”](#)
- Anweisungen in der Datendefinitionssprache (DDL): [Keyspaces](#)
- [Kontingente für Amazon Keyspaces \(für Apache Cassandra\)](#)

Wenn Sie einen Schlüsselraum erstellen, müssen Sie den Schlüsselraumnamen angeben.

Note

Die Replikationsstrategie des Schlüsselraums muss sein. `SingleRegionStrategy` repliziert Daten über drei Availability Zones in seiner. AWS-Region

Verwenden der Konsole

Um einen Schlüsselraum mit der Konsole zu erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Keyspaces aus.
3. Wählen Sie „Schlüsselraum erstellen“.
4. Geben Sie **myGSGKeyspace** im Feld „Schlüsselraumname“ den Namen für Ihren Schlüsselraum ein.

Einschränkungen für Namen:

- Darf nicht leer sein.
 - Zulässige Zeichen: alphanumerische Zeichen und Unterstrich ()_.
 - Die maximale Länge beträgt 48 Zeichen.
5. Um den Schlüsselraum zu erstellen, wählen Sie „Schlüsselraum erstellen“.
 6. Stellen Sie sicher, dass der Schlüsselraum erstellt myGSGKeyspace wurde, indem Sie wie folgt vorgehen:
 - a. Wählen Sie im Navigationsbereich Keyspaces aus.
 - b. Suchen Sie Ihren Schlüsselraum myGSGKeyspace in der Liste der Schlüsselräume.

Verwenden Sie CQL

Das folgende Verfahren erstellt mithilfe von CQL einen Schlüsselraum.

Um einen Schlüsselraum mit CQL zu erstellen


1. Öffnen Sie eine Befehlsshell und geben Sie Folgendes ein:

cqlsh

2. Erstellen Sie Ihren Keyspace mit dem folgenden CQL-Befehl.

```
CREATE KEYSPACE IF NOT EXISTS "myGSGKeyspace"  
  WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

`SingleRegionStrategy` verwendet einen Replikationsfaktor von drei und repliziert Daten in drei AWS Availability Zones in seiner Region.

 Note

Amazon Keyspaces verwendet standardmäßig alle Eingaben in Kleinbuchstaben, sofern Sie sie nicht in Anführungszeichen setzen. Beachten Sie in diesem Fall. "myGSGKeyspace"

3. Vergewissern Sie sich, dass Ihr Keyspace erstellt wurde.

```
SELECT * from system_schema.keyspaces ;
```

Ihr Schlüsselraum sollte aufgeführt sein.

Erstellen einer Tabelle

In einer Tabelle werden Ihre Daten organisiert und gespeichert. Der Primärschlüssel Ihrer Tabelle bestimmt, wie Daten in Ihrer Tabelle partitioniert werden. Der Primärschlüssel besteht aus einem erforderlichen Partitionsschlüssel und einer oder mehreren optionalen Clusterspalten. Die kombinierten Werte, aus denen sich der Primärschlüssel zusammensetzt, müssen für alle Daten der Tabelle eindeutig sein. Weitere Informationen zu Tabellen finden Sie in den folgenden Themen:

- Mit Tabellen arbeiten: [the section called “Erstellen von Tabellen”](#)
- DDL-Anweisungen: [Tabellen](#)
- Verwaltung von Tabellenressourcen: [Serverloses Ressourcenmanagement](#)
- Überwachung der Auslastung der Tabellenressourcen: [the section called “Überwachung mit CloudWatch”](#)
- [Kontingente für Amazon Keyspaces \(für Apache Cassandra\)](#)

Wenn Sie eine Tabelle erstellen, geben Sie Folgendes an:

- Der Name der Tabelle.
- Der Name und der Datentyp jeder Spalte in der Tabelle.
- Der Primärschlüssel für die Tabelle.

- Partitionsschlüssel — Erforderlich
- Spalten gruppieren — optional

Gehen Sie wie folgt vor, um eine Tabelle mit den angegebenen Spalten, Datentypen, Partitionsschlüsseln und Clusterspalten zu erstellen.

Verwenden der Konsole

Mit dem folgenden Verfahren wird die Tabelle `employees_tbl` mit diesen Spalten und Datentypen erstellt.

```
ID          text
name        text
region      text
division    text
project     text
role        text
pay_scale   int
vacation_hrs float
manager_id  text
```

Um eine Tabelle mit der Konsole zu erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Keyspaces aus.
3. Wählen Sie myGSGKeyspace den Schlüsselraum aus, in dem Sie diese Tabelle erstellen möchten.
4. Wählen Sie Create table (Tabelle erstellen) aus.
5. Geben Sie **employees_tbl** im Feld Tabellename einen Namen für Ihre Tabelle ein.

Einschränkungen für Namen:

- Darf nicht leer sein.
 - Zulässige Zeichen: alphanumerische Zeichen und Unterstrich (`_`).
 - Die maximale Länge beträgt 48 Zeichen.
6. Wiederholen Sie im Abschnitt Spalten die folgenden Schritte für jede Spalte, die Sie zu dieser Tabelle hinzufügen möchten.


Fügen Sie die folgenden Spalten und Datentypen hinzu.

```
id            text
name          text
region        text
division      text
project       text
role          text
pay_scale     int
vacation_hrs  float
manager_id    text
```

- a. Name — Geben Sie einen Namen für die Spalte ein.

Einschränkungen für Namen:
 - Darf nicht leer sein.
 - Zulässige Zeichen: alphanumerische Zeichen und Unterstrich (_).
 - Die maximale Länge beträgt 48 Zeichen.
 - b. Typ — Wählen Sie in der Liste der Datentypen den Datentyp für diese Spalte aus.
 - c. Wenn Sie eine weitere Spalte hinzufügen möchten, wählen Sie Spalte hinzufügen.
7. Wählen Sie unter Partitionsschlüssel die Option `id` als Partitionsschlüssel aus. Für jede Tabelle ist ein Partitionsschlüssel erforderlich. Ein Partitionsschlüssel kann aus einer oder mehreren Spalten bestehen.
 8. `division` als Clusterspalte hinzufügen. Clusterspalten sind optional und bestimmen die Sortierreihenfolge innerhalb jeder Partition.
 - a. Um eine Clusterspalte hinzuzufügen, wählen Sie Clustering-Spalte hinzufügen aus.
 - b. Wählen Sie in der Spaltenliste die Option `Division` aus. Wählen Sie in der Liste Reihenfolge die Option `ASC` aus, um die Werte in dieser Spalte in aufsteigender Reihenfolge zu sortieren. (Wählen Sie `DESC` für eine absteigende Reihenfolge.)
 9. Wählen Sie im Bereich Tabelleneinstellungen die Option Standardeinstellungen aus.
 10. Wählen Sie `Create table` (Tabelle erstellen) aus.
 11. Vergewissern Sie sich, dass Ihre Tabelle erstellt wurde.
 - a. Wählen Sie im Navigationsbereich `Tables` (Tabellen) aus.

- b. Vergewissern Sie sich, dass Ihre Tabelle in der Tabellenliste enthalten ist.
- c. Wählen Sie den Namen Ihrer Tabelle.
- d. Vergewissern Sie sich, dass alle Ihre Spalten und Datentypen korrekt sind.

 Note

Die Spalten werden möglicherweise nicht in der Reihenfolge aufgeführt, in der Sie sie der Tabelle hinzugefügt haben.

- e. Vergewissern Sie sich in der Cluster-Spalte, dass die Division mit „Wahr“ gekennzeichnet ist. Alle anderen Tabellenspalten sollten den Wert False haben.

Verwenden von CQL

Das folgende Verfahren erstellt mithilfe von CQL eine Tabelle mit den folgenden Spalten und Datentypen. Die `id` Spalte soll der Partitionsschlüssel sein.

```
id          text
name        text
region      text
division    text
project     text
role        text
pay_scale   int
vacation_hrs float
manager_id  text
```

Um eine Tabelle mit CQL zu erstellen

1. Öffnen Sie eine Befehlsshell und geben Sie Folgendes ein:


cqlsh

2. Geben Sie an der `cqlsh` Eingabeaufforderung (`cqlsh>`) einen Schlüsselraum an, in dem Ihre Tabelle erstellt werden soll.

```
USE "myGSGKeyspace" ;
```

3. Erstellen Sie an der Keyspace-Eingabeaufforderung (`cqlsh:keyspace_name>`) Ihre Tabelle, indem Sie den folgenden Code in Ihr Befehlsfenster eingeben.


```
CREATE TABLE IF NOT EXISTS "myGSGKeyspace".employees_tbl (  
  id text,  
  name text,  
  region text,  
  division text,  
  project text,  
  role text,  
  pay_scale int,  
  vacation_hrs float,  
  manager_id text,  
  PRIMARY KEY (id,division))  
WITH CLUSTERING ORDER BY (division ASC) ;
```

 Note

ASC ist die standardmäßige Clusterreihenfolge. Sie können auch eine DESC absteigende Reihenfolge angeben.

Beachten Sie, dass die `id` Spalte der Partitionsschlüssel sein soll. Dann `division` ist die Clusterspalte in aufsteigender Reihenfolge angeordnet (`ASC`).

4. Vergewissern Sie sich, dass Ihre Tabelle erstellt wurde.

```
SELECT * from system_schema.tables WHERE keyspace_name='myGSGKeyspace' ;
```

Ihre Tabelle sollte aufgeführt sein.

5. Überprüfen Sie die Struktur Ihrer Tabelle.

```
SELECT * FROM system_schema.columns WHERE keyspace_name = 'myGSGKeyspace' AND  
table_name = 'employees_tbl' ;
```

Vergewissern Sie sich, dass alle Spalten und Datentypen Ihren Erwartungen entsprechen. Die Reihenfolge der Spalten kann sich von der Reihenfolge in der `CREATE` Anweisung unterscheiden.

Um CRUD-Operationen (Create, Read, Update, Delete) mit den Daten in Ihrer Tabelle durchzuführen, fahren Sie mit fort [the section called “Schritt 2: CRUD-Operationen”](#).

Tutorial Schritt 2: Daten erstellen, lesen, aktualisieren und löschen (CRUD)

In diesem Abschnitt verwenden Sie den CQL-Editor auf der Konsole, um CRUD-Operationen (Create, Read, Update, Delete) mit den Daten in Ihrer Tabelle durchzuführen. Sie können die Befehle auch mit `cqlsh` ausführen.

Themen

- [Tutorial: Daten in eine Amazon Keyspaces-Tabelle einfügen und laden](#)
- [Tutorial: Aus einer Amazon Keyspaces-Tabelle lesen](#)
- [Tutorial: Daten in einer Amazon Keyspaces-Tabelle aktualisieren](#)
- [Tutorial: Daten in einer Amazon Keyspaces-Tabelle löschen](#)

Tutorial: Daten in eine Amazon Keyspaces-Tabelle einfügen und laden

Um Daten in Ihrer `employees_tbl` Tabelle zu erstellen, verwenden Sie die `INSERT` Anweisung, um eine einzelne Zeile hinzuzufügen.

1. Bevor Sie mit `cqlsh` Daten in Ihre Amazon Keyspaces-Tabelle schreiben können, müssen Sie die Schreibkonsistenz für die aktuelle `Cqlsh`-Sitzung auf einstellen. `LOCAL_QUORUM` Weitere Informationen zu den unterstützten Konsistenzstufen finden Sie unter [the section called “Konsistenzstufen schreiben”](#) Beachten Sie, dass dieser Schritt nicht erforderlich ist, wenn Sie den CQL-Editor in der AWS Management Console verwenden.

```
CONSISTENCY LOCAL_QUORUM;
```

2. Um einen einzelnen Datensatz einzufügen, führen Sie den folgenden Befehl im CQL-Editor aus.

```
INSERT INTO "myGSGKeyspace".employees_tbl (id, name, project, region, division,
role, pay_scale, vacation_hrs, manager_id)
VALUES ('012-34-5678', 'Russ', 'NightFlight', 'US', 'Engineering', 'IC', 3, 12.5,
'234-56-7890') ;
```

3. Stellen Sie sicher, dass die Daten korrekt zu Ihrer Tabelle hinzugefügt wurden, indem Sie den folgenden Befehl ausführen.

```
SELECT * FROM "myGSGKeyspace".employees_tbl ;
```

Um mehrere Datensätze aus einer Datei mit `cqlsh` einzufügen

1. [Laden Sie die Beispieldatendatei \(employees.csv\) herunter, die in der folgenden Archivdatei `sampledata.zip` enthalten ist.](#) Diese CSV-Datei (durch Kommas getrennte Werte) enthält die folgenden Daten. Merken Sie sich den Pfad, in dem Sie die Datei speichern.

ID	name	project	region	division	role	pay_scale	vacation_hrs	manager_id
123-45-6789	Bob	NightFlight	US	Engineering	Intern	1	0	234-56-7890
234-56-7890	Bob	NightFlight	US	Engineering	Manager	6	72	789-01-2345
345-67-8901	Sarah	Storm	US	Engineering	IC	4	108	234-56-7890
456-78-9012	Beth	NightFlight	US	Engineering	IC	7	100.5	234-56-7890
567-89-0123	Ahmed	NightFlight	US	Marketing	IC	4	88	678-90-1234
678-90-1234	Alan	Storm	US	Marketing	Manager	3	18.4	789-01-2345
789-01-2345	Roberta	All	US	Executive	CEO	15	184	None

2. Öffnen Sie eine Befehlsshell und geben Sie Folgendes ein:

`cqlsh`

3. Geben Sie an der `cqlsh` Eingabeaufforderung (`cqlsh>`) einen Schlüsselraum an.

```
USE "myGSGKeyspace" ;
```

4. Stellen Sie die Schreibkonsistenz auf ein. `LOCAL_QUORUM` Weitere Informationen zu den unterstützten Konsistenzstufen finden Sie unter [the section called "Konsistenzstufen schreiben"](#).

```
CONSISTENCY LOCAL_QUORUM;
```

5. Führen Sie an der Keyspace-Eingabeaufforderung (`cqlsh:keyspace_name>`) die folgende Abfrage aus.

```
COPY employees_tbl
(id,name,project,region,division,role,pay_scale,vacation_hrs,manager_id)
FROM 'path-to-the-csv-file/employees.csv' WITH delimiter=', ' AND header=TRUE ;
```

6. Stellen Sie sicher, dass die Daten korrekt zu Ihrer Tabelle hinzugefügt wurden, indem Sie die folgende Abfrage ausführen.

```
SELECT * FROM employees_tbl ;
```

Tutorial: Aus einer Amazon Keyspaces-Tabelle lesen

In [Tutorial: Daten in eine Amazon Keyspaces-Tabelle einfügen und laden](#) diesem Abschnitt haben Sie anhand der SELECT Anweisung überprüft, ob Sie Ihrer Tabelle erfolgreich Daten hinzugefügt haben. In diesem Abschnitt verfeinern Sie Ihre Verwendung von, SELECT um bestimmte Spalten und nur Zeilen anzuzeigen, die bestimmte Kriterien erfüllen.

Die allgemeine Form der SELECT Aussage lautet wie folgt.

```
SELECT column_list FROM table_name [WHERE condition [ALLOW FILTERING]] ;
```

Themen

- [Wählen Sie alle Daten in Ihrer Tabelle aus](#)
- [Auswahl einer Teilmenge von Spalten](#)
- [Auswahl einer Teilmenge von Zeilen](#)

Wählen Sie alle Daten in Ihrer Tabelle aus

Die einfachste Form der SELECT Anweisung gibt alle Daten in Ihrer Tabelle zurück.

Important

In einer Produktionsumgebung ist es in der Regel nicht empfehlenswert, diesen Befehl auszuführen, der alle Daten in Ihrer Tabelle zurückgibt.

Um alle Daten Ihrer Tabelle auszuwählen

- Führen Sie die folgende Abfrage aus.

```
SELECT * FROM "myGSGKeyspace".employees_tbl ;
```

Wenn Sie das Platzhalterzeichen (*) für verwenden, werden alle `column_list` Spalten ausgewählt.

Auswahl einer Teilmenge von Spalten

Um eine Teilmenge von Spalten abzufragen

- Führen Sie die folgende Abfrage aus `idname`, um nur die `manager_id` Spalten, und abzurufen.

```
SELECT name, id, manager_id FROM "myGSGKeyspace".employees_tbl ;
```

Die Ausgabe enthält nur die angegebenen Spalten in der Reihenfolge, die in der `SELECT` Anweisung aufgeführt ist.

Auswahl einer Teilmenge von Zeilen

Wenn Sie einen großen Datensatz abfragen, möchten Sie möglicherweise nur Datensätze, die bestimmte Kriterien erfüllen. Zu diesem Zweck können Sie eine `WHERE` Klausel an das Ende unserer `SELECT` Anweisung anhängen.

Um eine Teilmenge von Zeilen abzufragen

- Führen Sie die folgende Abfrage aus, um nur den Datensatz für den Mitarbeiter mit der ID `'234-56-7890'` abzurufen.

```
SELECT * FROM "myGSGKeyspace".employees_tbl WHERE id='234-56-7890' ;
```

Die vorherige `SELECT` Anweisung gibt nur die Zeilen zurück, in denen der `id` ist `234-56-7890`.

Die **WHERE** Klausel verstehen

Die `WHERE` Klausel wird verwendet, um die Daten zu filtern und nur die Daten zurückzugeben, die die angegebenen Kriterien erfüllen. Bei den angegebenen Kriterien kann es sich um eine einfache oder eine zusammengesetzte Bedingung handeln.

Wie verwendet man Bedingungen in einer **WHERE** Klausel

- Eine einfache Bedingung — Eine einzelne Spalte.

```
WHERE column_name=value
```

Sie können eine einfache Bedingung in einer WHERE Klausel verwenden, wenn eine der folgenden Bedingungen erfüllt ist:

- Die Spalte ist die einzige Spalte im Primärschlüssel der Tabelle.
- Sie fügen `ALLOW FILTERING` nach der Bedingung in der WHERE Klausel hinzu.

Beachten Sie, dass die Verwendung zu einer inkonsistenten Leistung führen `ALLOW FILTERING` kann, insbesondere bei großen und mehrfach partitionierten Tabellen.

- Eine zusammengesetzte Bedingung — Mehrere einfache Bedingungen, die miteinander verbunden sind. `AND`

```
WHERE column_name1=value1 AND column_name2=value2 AND column_name3=value3...
```

Sie können zusammengesetzte Bedingungen in einer WHERE Klausel verwenden, wenn eine der folgenden Bedingungen erfüllt ist:

- Die Spalten in der WHERE Klausel entsprechen exakt den Spalten im Primärschlüssel der Tabelle, nicht mehr und nicht weniger.
- Sie fügen `ALLOW FILTERING` nach der zusammengesetzten Bedingung in der WHERE Klausel hinzu, wie im folgenden Beispiel.

```
SELECT * FROM my_table WHERE col1=5 AND col2='Bob' ALLOW FILTERING ;
```

Beachten Sie, dass die Verwendung zu einer inkonsistenten Leistung führen `ALLOW FILTERING` kann, insbesondere bei großen und mehrfach partitionierten Tabellen.

Probieren Sie es aus

Erstellen Sie Ihre eigenen CQL-Abfragen, um Folgendes aus Ihrer `employees_tb1` Tabelle zu ermitteln:

- Finden Sie die `nameproject`, und `id` aller Mitarbeiter.
- Finden Sie heraus, an welchem Projekt Bob der Praktikant gerade arbeitet (geben Sie mindestens seinen Namen, sein Projekt und seine Rolle in der Ausgabe an).
- Fortgeschritten: Erstellen Sie eine Anwendung, um alle Mitarbeiter zu finden, die denselben Manager wie Bob der Praktikant haben. HINWEIS: Dies kann mehr als eine Abfrage erfordern.

- Fortgeschritten: Erstellen Sie eine Anwendung, um ausgewählte Spalten aller Mitarbeiter zu finden, die an dem Projekt arbeiten `NightFlight`. HINWEIS: Um dieses Problem zu lösen, sind möglicherweise mehrere Anweisungen erforderlich.

Tutorial: Daten in einer Amazon Keyspaces-Tabelle aktualisieren

Verwenden Sie die Anweisung, um die Daten in Ihrer `employees_tb1` Tabelle zu aktualisieren.

UPDATE

Die allgemeine Form der UPDATE Aussage lautet wie folgt.

```
UPDATE table_name SET column_name=new_value WHERE primary_key=value ;
```

Tip

- Sie können mehrere Spalten aktualisieren, indem Sie eine kommasetrennte Liste von Werten `column_names` und verwenden, wie im folgenden Beispiel.

```
UPDATE my_table SET col1='new_value_1', col2='new_value2' WHERE id='12345' ;
```

- Wenn der Primärschlüssel aus mehreren Spalten besteht, müssen alle Primärschlüsselspalten und ihre Werte in der `WHERE` Klausel enthalten sein.
- Sie können keine Spalte im Primärschlüssel aktualisieren, da dies den Primärschlüssel für den Datensatz ändern würde.

Um eine einzelne Zelle zu aktualisieren

Verwenden Sie Ihre `employees_tb1` Tabelle und geben Sie dem Mitarbeiter mit Personalausweis `567-89-0123` eine Gehaltserhöhung.

```
UPDATE "myGSGKeyspace".employees_tb1 SET pay_scale=5 WHERE id='567-89-0123' AND division='Marketing' ;
```

Vergewissern Sie sich, dass die Gehaltsskala des Mitarbeiters aktuell ist5.

```
SELECT * FROM "myGSGKeyspace".employees_tb1 WHERE id='567-89-0123' ;
```

Probieren Sie es aus

Fortgeschritten: Ihr Unternehmen hat Bob, den Praktikanten, eingestellt. Ändern Sie seinen Datensatz so, dass es seine Rolle ' IC ' und seine Gehaltsskala sind2.

Tutorial: Daten in einer Amazon Keyspaces-Tabelle löschen

Verwenden Sie die Anweisung, um Daten in Ihrer employees_tb1 Tabelle zu löschen. DELETE

Sie können Daten aus einer Zeile oder aus einer Partition löschen. Seien Sie vorsichtig beim Löschen von Daten, da Löschungen irreversibel sind.

Durch das Löschen einer oder aller Zeilen aus einer Tabelle wird die Tabelle nicht gelöscht. Somit können Sie es erneut mit Daten füllen. Durch das Löschen einer Tabelle werden die Tabelle und alle darin enthaltenen Daten gelöscht. Um die Tabelle wieder verwenden zu können, müssen Sie sie neu erstellen und ihr Daten hinzufügen. Durch das Löschen eines Schlüsselraums werden der Schlüsselraum und alle darin enthaltenen Tabellen gelöscht. Um den Schlüsselraum und die Tabellen verwenden zu können, müssen Sie sie neu erstellen und sie dann mit Daten füllen.

Zellen löschen

Beim Löschen einer Spalte aus einer Zeile werden die Daten aus der angegebenen Zelle entfernt. Wenn Sie diese Spalte mithilfe einer SELECT Anweisung anzeigen, werden die Daten als *Null* angezeigt, obwohl an dieser Stelle kein Nullwert gespeichert wird.

Die allgemeine Syntax zum Löschen einer oder mehrerer bestimmter Spalten lautet wie folgt.

```
DELETE column_name1[, column_name2...] FROM table_name WHERE condition ;
```

In Ihrer employees_tb1 Tabelle können Sie sehen, dass der CEO "None" für einen Manager zuständig ist. Löschen Sie zunächst diese Zelle, sodass keine Daten darin enthalten sind.

Um eine bestimmte Zelle zu löschen

1. Führen Sie die folgende DELETE Abfrage aus.

```
DELETE manager_id FROM "myGSGKeyspace".employees_tb1 WHERE id='789-01-2345' AND division='Executive';
```

2. Stellen Sie sicher, dass der Löschvorgang wie erwartet durchgeführt wurde.


```
SELECT * FROM "myGSGKeyspace".employees_tbl WHERE id='789-01-2345' AND
division='Executive';
```

Zeilen löschen

Es kann vorkommen, dass Sie eine ganze Zeile löschen müssen, z. B. wenn ein Mitarbeiter in den Ruhestand geht. Die allgemeine Syntax für das Löschen einer Zeile lautet wie folgt.

```
DELETE FROM table_name WHERE condition ;
```

Um eine Zeile zu löschen

1. Führen Sie die folgende DELETE Abfrage aus.

```
DELETE FROM "myGSGKeyspace".employees_tbl WHERE id='456-78-9012' AND
division='Engineering';
```

2. Stellen Sie sicher, dass der Löschvorgang wie erwartet durchgeführt wurde.

```
SELECT * FROM "myGSGKeyspace".employees_tbl WHERE id='456-78-9012' AND
division='Engineering';
```

Tutorial Schritt 3: Löschen Sie eine Tabelle und einen Schlüsselraum in Amazon Keyspaces

Um zu vermeiden, dass Ihnen für Tabellen und Daten, die Sie nicht benötigen, Gebühren berechnet werden, löschen Sie alle Tabellen und Schlüsselbereiche, die Sie nicht verwenden. Wenn Sie eine Tabelle löschen, werden die Tabelle und ihre Daten gelöscht, und es fallen keine Gebühren mehr für sie an. Der Schlüsselraum bleibt jedoch bestehen. Wenn Sie einen Schlüsselraum löschen, werden der Schlüsselraum und alle zugehörigen Tabellen gelöscht, und es fallen keine Gebühren mehr für ihn an.

Löschen einer Tabelle

Sie können eine Tabelle mit der Konsole oder CQL löschen. Wenn Sie eine Tabelle löschen, werden die Tabelle und alle ihre Daten gelöscht.

Verwenden der Konsole

Das folgende Verfahren löscht eine Tabelle und alle ihre Daten mithilfe von AWS Management Console

Um eine Tabelle mit der Konsole zu löschen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tables (Tabellen) aus.
3. Wählen Sie das Feld links neben dem Namen jeder Tabelle aus, die Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Geben Sie auf dem Bildschirm Tabelle löschen den Text **Delete** in das Feld ein. Wählen Sie dann Tabelle löschen.
6. Um zu überprüfen, ob die Tabelle gelöscht wurde, wählen Sie im Navigationsbereich Tabellen aus und vergewissern Sie sich, dass die `employees_tbl` Tabelle nicht mehr aufgeführt ist.

Verwenden von CQL

Das folgende Verfahren löscht eine Tabelle und all ihre Daten mithilfe von CQL.

Um eine Tabelle mit CQL zu löschen

1. Öffnen Sie eine Befehlsshell und geben Sie Folgendes ein:

cqlsh

2. Löschen Sie Ihre Tabelle, indem Sie an der Keyspace-Eingabeaufforderung den folgenden Befehl eingeben (`cqlsh:keyspace_name>`).

```
DROP TABLE IF EXISTS "myGSGKeyspace".employees_tbl ;
```

3. Vergewissern Sie sich, dass Ihre Tabelle gelöscht wurde.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = 'myGSGKeyspace' ;
```

Ihre Tabelle sollte nicht aufgeführt werden.

Einen Schlüsselraum löschen

Sie können einen Schlüsselraum entweder mit der AWS Management Console oder mit der CQL löschen. Wenn Sie einen Schlüsselraum löschen, werden der Schlüsselraum und alle zugehörigen Tabellen und Daten gelöscht.

Verwenden des AWS Management Console

Das folgende Verfahren löscht einen Schlüsselraum und alle zugehörigen Tabellen und Daten mithilfe von AWS Management Console.

Um einen Schlüsselraum mit der Konsole zu löschen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Keyspaces aus.
3. Wählen Sie das Feld links neben dem Namen jedes Schlüsselraums aus, den Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Geben Sie auf dem Bildschirm „Schlüsselraum löschen“ **Delete** in das Feld ein. Wählen Sie dann „Schlüsselraum löschen“.
6. Um zu überprüfen, ob der Schlüsselraum gelöscht wurde, wählen Sie im Navigationsbereich Keyspaces aus und vergewissern Sie sich, dass er nicht mehr aufgeführt ist. Da Sie den Schlüsselraum gelöscht haben, sollte die `employees_tbl` Tabelle unter Tabellen ebenfalls nicht aufgeführt werden.

Verwenden von CQL

Das folgende Verfahren löscht einen Schlüsselraum und all seine Tabellen und Daten mithilfe von CQL.

Um einen Schlüsselraum mit CQL zu löschen

1. Öffnen Sie eine Befehlsshell und geben Sie Folgendes ein:

```
cqlsh
```

2. Löschen Sie Ihren Keyspace, indem Sie an der Keyspace-Eingabeaufforderung den folgenden Befehl eingeben `()cqlsh: keyspace_name>`.

```
DROP KEYSPACE IF EXISTS "myGSGKeyspace" ;
```

3. Stellen Sie sicher, dass Ihr Schlüsselraum gelöscht wurde.

```
SELECT * from system_schema.keyspaces ;
```

Ihr Keyspace sollte nicht aufgeführt werden. Beachten Sie, dass es zu Verzögerungen kommen kann, bis der Schlüsselraum gelöscht wird, da es sich um einen asynchronen Vorgang handelt.

Migration zu Amazon Keyspaces

Die Migration zu Amazon Keyspaces (für Apache Cassandra) bietet eine Reihe überzeugender Vorteile für Unternehmen und Organisationen. Hier sind einige wichtige Vorteile, die Amazon Keyspaces zu einer attraktiven Wahl für die Migration machen.

- **Skalierbarkeit** — Amazon Keyspaces ist darauf ausgelegt, enorme Workloads zu bewältigen und sich nahtlos skalieren zu lassen, um wachsenden Datenmengen und Datenströmen gerecht zu werden. Bei herkömmlichem Cassandra erfolgt die Skalierung nicht bei Bedarf und erfordert die Planung zukünftiger Spitzenwerte. Mit Amazon Keyspaces können Sie Ihre Tabellen je nach Bedarf einfach nach oben oder unten skalieren und so sicherstellen, dass Ihre Anwendungen plötzliche Datenverkehrsspitzen ohne Leistungseinbußen bewältigen können.
- **Leistung** — Amazon Keyspaces bietet Datenzugriff mit niedriger Latenz, sodass Anwendungen Daten mit außergewöhnlicher Geschwindigkeit abrufen und verarbeiten können. Die verteilte Architektur stellt sicher, dass Lese- und Schreibvorgänge auf mehrere Knoten verteilt werden, sodass selbst bei hohen Anforderungsraten konsistente Antwortzeiten im einstelligen Millisekundenbereich erzielt werden.
- **Vollständig verwaltet** — Amazon Keyspaces ist ein vollständig verwalteter Service von AWS. Das bedeutet, dass er sich um die betrieblichen Aspekte der Datenbankverwaltung AWS kümmert, einschließlich Bereitstellung, Konfiguration, Patching, Backups und Skalierung. Auf diese Weise können Sie sich mehr auf die Entwicklung Ihrer Anwendungen und weniger auf Datenbankverwaltungsaufgaben konzentrieren.
- **Serverlose Architektur** — Amazon Keyspaces ist serverlos. Sie zahlen nur für die verbrauchte Kapazität, ohne dass eine Vorauszahlung der Kapazität erforderlich ist. Sie müssen weder Server verwalten noch Instanzen auswählen. Dieses pay-per-request Modell bietet Kosteneffizienz und minimalen Betriebsaufwand, da Sie nur für die Ressourcen zahlen, die Sie verbrauchen, ohne dass Kapazitäten bereitgestellt und überwacht werden müssen.
- **NoSQL-Flexibilität mit Schema** — Amazon Keyspaces folgt einem NoSQL-Datenmodell, das Flexibilität beim Schemadesign bietet. Mit Amazon Keyspaces können Sie strukturierte, halbstrukturierte und unstrukturierte Daten speichern, sodass sie sich gut für den Umgang mit unterschiedlichen und sich weiterentwickelnden Datentypen eignen. Darüber hinaus führt Amazon Keyspaces beim Schreiben eine Schemavalidierung durch, was eine zentralisierte Weiterentwicklung des Datenmodells ermöglicht. Diese Flexibilität ermöglicht schnellere Entwicklungszyklen und eine einfachere Anpassung an sich ändernde Geschäftsanforderungen.

- Hohe Verfügbarkeit und Beständigkeit — Amazon Keyspaces repliziert Daten über mehrere [Availability Zones](#) innerhalb einer und gewährleistet so eine AWS-Region hohe Verfügbarkeit und Datenbeständigkeit. Es kümmert sich automatisch um Replikation, Failover und Wiederherstellung und minimiert so das Risiko von Datenverlusten oder Serviceunterbrechungen. Amazon Keyspaces bietet eine Verfügbarkeits-SLA von bis zu 99,999% [Für noch mehr Stabilität und lokale Lesevorgänge mit geringer Latenz bietet Amazon Keyspaces Replikation in mehreren Regionen.](#)
- Sicherheit und Compliance — Amazon Keyspaces lässt sich AWS Identity and Access Management für eine differenzierte Zugriffskontrolle integrieren. Es bietet Verschlüsselung im Ruhezustand und bei der Übertragung und trägt so zur Verbesserung der Sicherheit Ihrer Daten bei. Amazon Keyspaces erfüllt außerdem Sicherheitsstandards und Datenschutzgesetze, einschließlich HIPAA, PCI DSS und GDPR, sodass Sie die gesetzlichen Anforderungen erfüllen können.
- Integration mit dem AWS Ökosystem — Als Teil des AWS Ökosystems lässt sich Amazon Keyspaces nahtlos in andere AWS-Services Systeme integrieren AWS CloudFormation, beispielsweise Amazon CloudWatch und AWS CloudTrail. Diese Integration ermöglicht es Ihnen, serverlose Architekturen zu erstellen, Infrastruktur als Code zu nutzen und datengesteuerte Echtzeitanwendungen zu erstellen.

Allgemeine Überlegungen zu Migrationen zu Amazon Keyspaces

- Teilen Sie die Migration in kleinere Komponenten auf.

Betrachten Sie die folgenden Migrationseinheiten und ihren potenziellen Platzbedarf in Bezug auf die Rohdatengröße. Die Migration kleinerer Datenmengen in einer oder mehreren Phasen kann dazu beitragen, Ihre Migration zu vereinfachen.

- Nach Clustern — Migrieren Sie alle Ihre Cassandra-Daten auf einmal. Dieser Ansatz kann für kleinere Cluster in Ordnung sein.
- Nach Schlüsselraum oder Tabelle — Teilen Sie Ihre Migration in Gruppen von Schlüsselräumen oder Tabellen auf. Dieser Ansatz kann Ihnen dabei helfen, Daten in Phasen zu migrieren, die Ihren Anforderungen für jeden Workload entsprechen.
- Nach Daten — Erwägen Sie die Migration von Daten für eine bestimmte Gruppe von Benutzern oder Produkten, um die Datenmenge noch weiter zu reduzieren.
- Priorisieren Sie anhand der Einfachheit, welche Daten zuerst migriert werden sollen.

Überlegen Sie, ob Sie Daten haben, die zunächst einfacher migriert werden könnten, z. B. Daten, die sich zu bestimmten Zeiten nicht ändern, Daten aus nächtlichen Batch-Jobs, Daten, die während der Offline-Zeiten nicht verwendet werden, oder Daten aus internen Apps.

Themen

- [Anleitung für die Migration von Daten aus Apache Cassandra](#)
- [Tools für die Migration von Daten zu Amazon Keyspaces](#)

Anleitung für die Migration von Daten aus Apache Cassandra

Für eine erfolgreiche Migration von Apache Cassandra zu Amazon Keyspaces empfehlen wir eine sorgfältige Planung und einen Vergleich der verfügbaren Optionen. In diesem Thema wird beschrieben, wie der Migrationsprozess funktioniert, welche Tools verfügbar sind und wie Sie verschiedene Migrationsstrategien bewerten können, um die auszuwählen, die Ihren Anforderungen am besten entspricht.

Themen

- [Funktionale Kompatibilität](#)
- [Schätzen Sie die Preise für Amazon Keyspaces](#)
- [Wählen Sie eine Migrationsstrategie](#)
- [Offline-Migration zu Amazon Keyspaces](#)

Funktionale Kompatibilität

Überlegen Sie sich vor der Migration sorgfältig die funktionalen Unterschiede zwischen Apache Cassandra und Amazon Keyspaces. Amazon Keyspaces unterstützt alle häufig verwendeten Cassandra-Datenebenenoperationen, z. B. das Erstellen von Schlüsselräumen und Tabellen, das Lesen von Daten und das Schreiben von Daten. Es gibt jedoch einige Cassandra-APIs, die Amazon Keyspaces nicht unterstützt. Weitere Informationen zu unterstützten APIs finden Sie unter [the section called “Unterstützte Cassandra-APIs, Operationen, Funktionen und Datentypen”](#) Einen Überblick über alle funktionalen Unterschiede zwischen Amazon Keyspaces und Apache Cassandra finden Sie unter [the section called “Funktionelle Unterschiede zu Apache Cassandra”](#)

Um die Cassandra-APIs und das von Ihnen verwendete Schema mit den unterstützten Funktionen in Amazon Keyspaces zu vergleichen, können Sie ein Kompatibilitätsskript ausführen, das im Amazon Keyspaces-Toolkit verfügbar ist. [GitHub](#)

Wie benutzt man das Kompatibilitätsskript

1. Laden Sie das Kompatibilitäts-Python-Skript von [herunter GitHub](#) und verschieben Sie es an einen Speicherort, der Zugriff auf Ihren vorhandenen Apache Cassandra-Cluster hat.
2. Das Kompatibilitätsskript verwendet ähnliche Parameter wie CQLSH. `--port` Geben Sie für `--host` und geben Sie die IP-Adresse und den Port ein, mit dem Sie eine Verbindung herstellen und Abfragen zu einem der Cassandra-Knoten in Ihrem Cluster ausführen. Wenn Ihr Cassandra-Cluster Authentifizierung verwendet, müssen Sie auch `-username` und `-password` angeben. Um das Kompatibilitätsskript auszuführen, können Sie den folgenden Befehl verwenden.

```
python toolkit-compat-tool.py --host hostname or IP -u "username" -p "password" --port native transport port
```

Schätzen Sie die Preise für Amazon Keyspaces

Dieser Abschnitt bietet einen Überblick über die Informationen, die Sie aus Ihren Apache Cassandra-Tabellen sammeln müssen, um die geschätzten Kosten für Amazon Keyspaces zu berechnen. Jede Ihrer Tabellen benötigt unterschiedliche Datentypen, muss unterschiedliche CQL-Abfragen unterstützen und unterhält einen eigenen Lese-/Schreibverkehr. Wenn Sie Ihre Anforderungen auf der Grundlage von Tabellen berücksichtigen, entspricht dies den Amazon Keyspaces-Modi für die Ressourcenisolierung auf Tabellenebene und die Kapazitätsmodi für den [Lese-/Schreibdurchsatz](#). Mit Amazon Keyspaces können Sie Lese-/Schreibkapazitäten und [automatische Skalierungsrichtlinien](#) für Tabellen unabhängig voneinander definieren. Wenn Sie die Tabellenanforderungen verstehen, können Sie Tabellen für die Migration auf der Grundlage von Funktionalität, Kosten und Migrationsaufwand priorisieren.

Erfassen Sie vor einer Migration die folgenden Cassandra-Tabellenmetriken. Diese Informationen helfen Ihnen dabei, die Kosten Ihres Workloads auf Amazon Keyspaces abzuschätzen.

- **Tabellenname** — Der Name des vollqualifizierten Schlüsselraums und der Tabellenname.
- **Beschreibung** — Eine Beschreibung der Tabelle, z. B. wie sie verwendet wird oder welche Art von Daten darin gespeichert sind.

- Durchschnittliche Lesevorgänge pro Sekunde — Die durchschnittliche Anzahl von Lesevorgängen in der Tabelle auf Koordinatenebene über ein großes Zeitintervall.
- Durchschnittliche Schreibvorgänge pro Sekunde — Die durchschnittliche Anzahl von Schreibvorgängen in der Tabelle auf Koordinatenebene über einen großen Zeitraum.
- Durchschnittliche Zeilengröße in Byte — Die durchschnittliche Zeilengröße in Byte.
- Speichergröße in GB — Die ungefähre Speichergröße für eine Tabelle.
- Aufschlüsselung der Lesekonsistenz — Der Prozentsatz der Lesevorgänge, bei denen letztendlich Konsistenz (LOCAL_ONEoderONE) verwendet wird, im Vergleich zu starker Konsistenz (LOCAL_QUORUM).

Diese Tabelle zeigt ein Beispiel für die Informationen zu Ihren Tabellen, die Sie bei der Planung einer Migration zusammenstellen müssen.

Tabellenn ame	Beschreib ung	Durchschn ittliche Lesevorgä nge pro Sekunde	Durchschn ittliche Schreibvo rgänge pro Sekunde	Durchschn ittliche Zeilengrö ße in Byte	Speicherg röße in GB	Lesen Sie die Konsisten zübersicht
mykeyspac e.mytable	Wird zum Speichern des Warenkorb verlaufs verwendet	10.000	5,000	2.200	2.000	100% LOCAL_ONE
mein Schlüssel raum. Meine Tabelle 2	Wird verwendet , um die neuesten Profilinf ormatione n zu speichern	20 000	1.000	850	1.000	25% LOCAL_QUO RUM 75% LOCAL_ONE

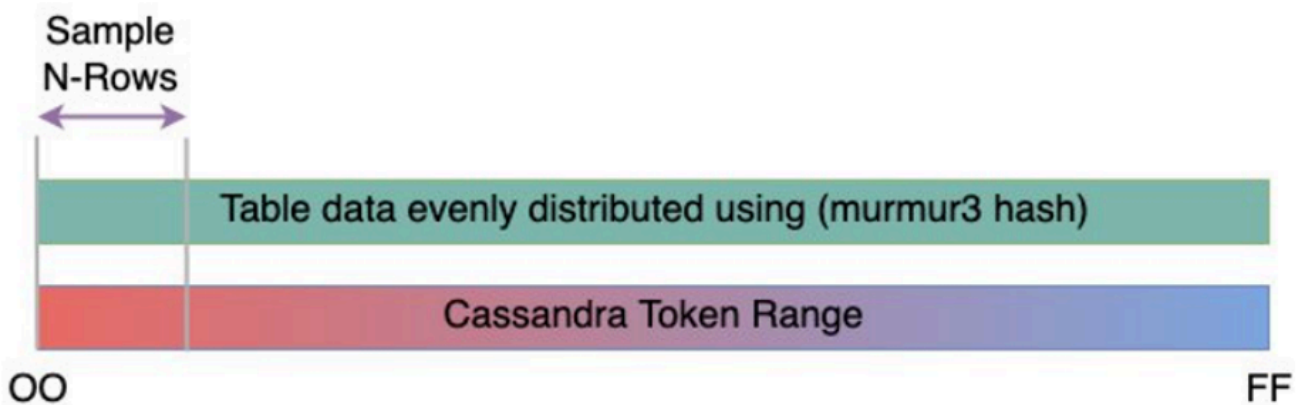
Wie erfasst man Tabellenmetriken

Dieser Abschnitt enthält schrittweise Anweisungen zum Sammeln der erforderlichen Tabellenmetriken aus Ihrem vorhandenen Cassandra-Cluster. Zu diesen Metriken gehören Zeilengröße, Tabellengröße und Lese-/Schreibanforderungen pro Sekunde (RPS). Sie ermöglichen es Ihnen, die Durchsatzkapazitätsanforderungen für eine Amazon Keyspaces-Tabelle zu bewerten und die Preise zu schätzen.

Wie erfasst man Tabellenmetriken in der Cassandra-Quelltabelle

1. Ermitteln Sie die Zeilengröße

Die Zeilengröße ist wichtig für die Bestimmung der Lese- und Schreibkapazitätsauslastung in Amazon Keyspaces. Das folgende Diagramm zeigt die typische Datenverteilung über einen Cassandra-Token-Bereich.



Sie können ein auf verfügbares Sampler-Skript zur Zeilengröße verwenden, um Metriken zur Zeilengröße für jede Tabelle in Ihrem Cassandra-Cluster [GitHub](#) zu sammeln. Das Skript exportiert Tabellendaten aus Apache Cassandra, indem es die Mindest `cqlsh -awk`, Höchst-, Durchschnitts- und Standardabweichung der Zeilengröße anhand eines konfigurierbaren Stichprobensatzes von Tabellendaten verwendet und berechnet. Der Zeilengrößen-Sampler übergibt die Argumente `ancqlsh`, sodass dieselben Parameter verwendet werden können, um eine Verbindung herzustellen und aus Ihrem Cassandra-Cluster zu lesen.

Die folgende Aussage ist ein Beispiel dafür.

```
./row-size-sampler.sh 10.22.33.44 9142 \\
-u "username" -p "password" --ssl
```

Weitere Informationen zur Berechnung der Zeilengröße in Amazon Keyspaces finden Sie unter [the section called “Berechnung der Zeilengröße”](#).

2. Ermitteln Sie die Tabellengröße

Mit Amazon Keyspaces müssen Sie Speicher nicht im Voraus bereitstellen. Amazon Keyspaces überwacht kontinuierlich die fakturierbare Größe Ihrer Tabellen, um Ihre Speichergebühren zu ermitteln. Speicherplatz wird pro GB-Monat abgerechnet. Die Tabellengröße von Amazon Keyspaces basiert auf der Rohgröße (unkomprimiert) eines einzelnen Replikats. Um die Tabellengröße in Amazon Keyspaces zu überwachen, können Sie die Metrik `BillableTableSizeInBytes`, die für jede Tabelle in der AWS Management Console angezeigt wird.

Um die abrechnungsfähige Größe Ihrer Amazon Keyspaces-Tabelle zu schätzen, können Sie eine der beiden folgenden Methoden verwenden:

- Verwenden Sie die durchschnittliche Zeilengröße und multiplizieren Sie sie mit der Anzahl oder den Zeilen.

Sie können die Größe der Amazon Keyspaces-Tabelle schätzen, indem Sie die durchschnittliche Zeilengröße mit der Anzahl der Zeilen aus Ihrer Cassandra-Quelltabelle multiplizieren. Verwenden Sie das Beispielskript für die Zeilengröße aus dem vorherigen Abschnitt, um die durchschnittliche Zeilengröße zu ermitteln. Um die Zeilenanzahl zu erfassen, können Sie Tools wie `dsbulk count` die Bestimmung der Gesamtzahl der Zeilen in Ihrer Quelltabelle verwenden.

- Verwenden Sie `dienodetool`, um Tabellenmetadaten zu sammeln.

`Nodetool` ist ein in der Apache Cassandra-Distribution enthaltenes Verwaltungstool, das Einblick in den Status des Cassandra-Prozesses bietet und Tabellenmetadaten zurückgibt. Sie können `nodetool` verwenden, um Metadaten zur Tabellengröße als Stichprobe zu verwenden und damit die Tabellengröße in Amazon Keyspaces zu extrapolieren. Der zu verwendende Befehl lautet `nodetool tablestats`. `Tablestats` gibt die Größe und das Komprimierungsverhältnis der Tabelle zurück. Die Größe der Tabelle wird wie `tablelivespace` für die Tabelle gespeichert und Sie können sie durch die `compression ratio` teilen. Dann multiplizieren Sie diesen Größenwert mit der Anzahl der Knoten. Schließlich dividieren Sie durch den Replikationsfaktor (normalerweise drei). Dies ist die vollständige Formel für die Berechnung, anhand derer Sie die Tabellengröße beurteilen können.

```
((tablelivespace / compression ratio) * (total number of nodes)) / (replication factor)
```

Nehmen wir an, dass Ihr Cassandra-Cluster 12 Knoten hat. Die Ausführung des `nodetool tablestats` Befehls gibt a `tablelivespace` von 200 GB und a `compression ratio` von 0,5 zurück. Der Schlüsselraum hat einen Replikationsfaktor von drei. So sieht die Berechnung für dieses Beispiel aus.

```
(200 GB / 0.5) * (12 nodes) / (replication factor of 3)
= 4,800 GB / 3
= 1,600 GB is the table size estimate for Amazon
Keyspaces
```

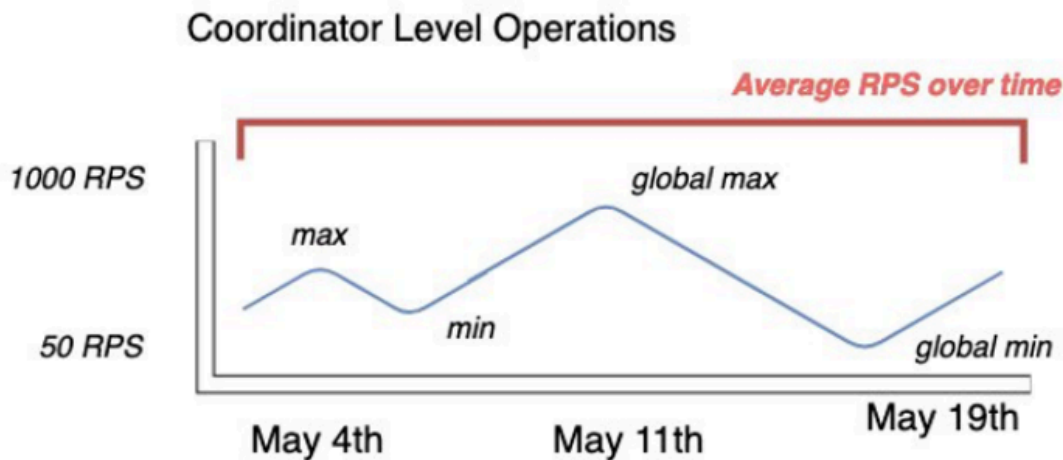
3. Erfassen Sie die Anzahl der Lese- und Schreibvorgänge

Um die Kapazitäts- und Skalierungsanforderungen für Ihre Amazon Keyspaces-Tabellen zu ermitteln, erfassen Sie vor der Migration die Lese- und Schreibanzforderungsrate Ihrer Cassandra-Tabellen.

Amazon Keyspaces ist serverlos und Sie zahlen nur für das, was Sie nutzen. Im Allgemeinen richtet sich der Preis für den Lese-/Schreibdurchsatz in Amazon Keyspaces nach der Anzahl und Größe der Anfragen. In Amazon Keyspaces gibt es zwei Kapazitätsmodi: den [On-Demand-Kapazitätsmodus](#) und den [Bereitstellungsmodus](#). Der On-Demand-Kapazitätsmodus ist eine flexible Abrechnungsoption, mit der Tausende von Anfragen pro Sekunde bearbeitet werden können, ohne dass eine Kapazitätsplanung erforderlich ist. Diese Option bietet pay-per-request Preise für Lese- und Schreibfragen, sodass Sie nur für das bezahlen, was Sie tatsächlich nutzen. Wenn Sie den Modus Bereitgestellte Durchsatzkapazität wählen, geben Sie die Anzahl der Lese- und Schreibvorgänge pro Sekunde an, die für Ihre Anwendung erforderlich sind. Auf diese Weise können Sie Ihre Amazon Keyspaces-Nutzung so verwalten, dass sie bei oder unter einer bestimmten Anforderungsrate bleibt, um den Preis zu optimieren und die Planbarkeit aufrechtzuerhalten. Der Bereitstellungsmodus bietet [auto Skalierung](#), um Ihre bereitgestellte Rate automatisch an die Skalierung nach oben oder unten anzupassen, um die betriebliche Effizienz zu verbessern. Weitere Informationen zur serverlosen Ressourcenverwaltung finden Sie unter [Serverloses Ressourcenmanagement](#)

Da Sie Lese- und Schreibdurchsatzkapazität in Amazon Keyspaces separat bereitstellen, müssen Sie die Anforderungsrate für Lese- und Schreibvorgänge in Ihren vorhandenen Tabellen unabhängig voneinander messen.

Um die genauesten Nutzungsmetriken aus Ihrem bestehenden Cassandra-Cluster zu sammeln, erfassen Sie die durchschnittlichen Anfragen pro Sekunde (RPS) für Lese- und Schreibvorgänge auf Koordinatorebene über einen längeren Zeitraum für eine Tabelle, die über alle Knoten in einem einzigen Rechenzentrum aggregiert wird. Durch die Erfassung des durchschnittlichen RPS über einen Zeitraum von mindestens mehreren Wochen werden Spitzen und Täler in Ihren Datenverkehrsmustern erfasst, wie in der folgenden Abbildung dargestellt.



Sie haben zwei Möglichkeiten, die Lese- und Schreibanzahl Ihrer Cassandra-Tabelle zu bestimmen.

- Verwenden Sie das bestehende Cassandra-Monitoring

Sie können die in der folgenden Tabelle aufgeführten Metriken verwenden, um Lese- und Schreibanzahl zu beobachten. Beachten Sie, dass sich die Namen der Metriken je nach dem von Ihnen verwendeten Überwachungstool ändern können.

Dimension	Cassandra JMX-Metrik
Schreibt	<code>org.apache.cassandra.metrics:type=ClientRequest,scope=Write,name=Latency#Count</code>

Dimension	Cassandra JMX-Metrik
Liest	<code>org.apache.cassandra.metrics:type=ClientRequest,scope=Read,name=Latency#Count</code>

- Verwenden der `nodetool`

Verwenden Sie `nodetool tablestats` und `nodetool info`, um durchschnittliche Lese- und Schreibvorgänge aus der Tabelle zu erfassen. `tablestats` gibt die Gesamtzahl der Lese- und Schreibvorgänge ab dem Zeitpunkt zurück, zu dem der Knoten initiiert wurde. `nodetool info` gibt die Betriebszeit für einen Knoten in Sekunden an. Um den Durchschnitt der Lese- und Schreibvorgänge pro Sekunde zu ermitteln, dividieren Sie die Anzahl der Lese- und Schreibvorgänge durch die Betriebszeit des Knotens in Sekunden. Dann dividieren Sie für Lesevorgänge durch den Konsistenzgrad und für Schreibvorgänge dividieren Sie durch den Replikationsfaktor. Diese Berechnungen werden in den folgenden Formeln ausgedrückt.

Formel für durchschnittliche Lesevorgänge pro Sekunde:

```
((number of reads * number of nodes in cluster) / read consistency quorum
(2)) / uptime
```

Formel für durchschnittliche Schreibvorgänge pro Sekunde:

```
((number of writes * number of nodes in cluster) / replication factor of 3) /
uptime
```

Nehmen wir an, wir haben einen 12-Knoten-Cluster, der seit 4 Wochen aktiv ist. `nodetool info` gibt 2.419.200 Sekunden Betriebszeit zurück und `nodetool tablestats` gibt 1 Milliarde Schreibvorgänge und 2 Milliarden Lesevorgänge zurück. Dieses Beispiel würde zu der folgenden Berechnung führen.

```
((2 billion reads * 12 in cluster) / read consistency quorum (2)) / 2,419,200
seconds
= 12 billion reads / 2,419,200 seconds
= 4,960 read request per second
((1 billion writes * 12 in cluster) / replication
factor of 3) / 2,419,200 seconds
```

```
= 4 billion writes / 2,419,200 seconds  
= 1,653 write request per second
```

4. Ermitteln Sie die Kapazitätsauslastung der Tabelle

Um die durchschnittliche Kapazitätsauslastung zu schätzen, beginnen Sie mit den durchschnittlichen Anforderungsraten und der durchschnittlichen Zeilengröße Ihrer Cassandra-Quelltabelle.

Amazon Keyspaces verwendet Lesekapazitätseinheiten (RCUs) und Schreibkapazitätseinheiten (WCUs), um die bereitgestellte Durchsatzkapazität für Lese- und Schreibvorgänge für Tabellen zu messen. Für diese Schätzung verwenden wir diese Einheiten, um den Lese- und Schreibkapazitätsbedarf der neuen Amazon Keyspaces-Tabelle nach der Migration zu berechnen. Später in diesem Thema werden wir erörtern, wie sich die Wahl zwischen Bereitstellungs- und On-Demand-Kapazitätsmodus auf die Abrechnung auswirkt. Für die Schätzung der Kapazitätsauslastung gehen wir jedoch davon aus, dass sich die Tabelle im Bereitstellungsmodus befindet.

Eine RCU steht für eine LOCAL_QUORUM oder zwei LOCAL_ONE Leseanforderungen für eine Zeile mit einer Größe von bis zu 4 KB. Wenn Sie eine Zeile lesen müssen, die größer als 4 KB ist, verwendet der Lesevorgang zusätzliche RCUs. Die Gesamtzahl der erforderlichen RCUs hängt von der Zeilengröße und davon ab, ob Sie Konsistenz verwenden LOCAL_QUORUM oder LOCAL_ONE lesen möchten. Zum Lesen einer 8-KB-Zeile sind beispielsweise 2 RCUs mit LOCAL_QUORUM Lesekonsistenz und 1 RCU erforderlich, wenn Sie Lesekonsistenz wählen LOCAL_ONE.

Eine WCU entspricht einem Schreibvorgang für eine Zeile mit einer Größe von bis zu 1 KB. Bei allen Schreibvorgängen wird LOCAL_QUORUM Konsistenz verwendet, und es fallen keine zusätzlichen Gebühren für die Verwendung von Lightweight Transactions (LWTs) an. Wenn Sie eine Zeile schreiben müssen, die größer als 1 KB ist, verwendet der Schreibvorgang zusätzliche WCUs. Die Gesamtzahl der benötigten WCUs hängt von der Zeilengröße ab. Wenn Ihre Zeilengröße beispielsweise 2 KB beträgt, benötigen Sie 2 WCUs, um eine Schreibanforderung auszuführen.

Die folgende Formel kann verwendet werden, um die erforderlichen RCUs und WCUs abzuschätzen. Die Lesekapazität in RCUs kann bestimmt werden, indem Lesevorgänge pro Sekunde mit der Anzahl der pro Lesevorgang gelesenen Zeilen multipliziert werden, multipliziert mit der durchschnittlichen Zeilengröße geteilt durch 4 KB und aufgerundet auf die nächste ganze Zahl.

Die Schreibkapazität in WCUs kann bestimmt werden, indem die Anzahl der Anfragen mit der durchschnittlichen Zeilengröße geteilt durch 1 KB multipliziert und auf die nächste ganze Zahl aufgerundet wird. Dies wird in den folgenden Formeln ausgedrückt.

$$\text{Read requests per second} * \text{ROUNDUP}(\text{Average Row Size}/4096 \text{ per unit}) = \text{RCUs per second}$$

$$\text{Write requests per second} * \text{ROUNDUP}(\text{Average Row Size}/1024 \text{ per unit}) = \text{WCUs per second}$$

Wenn Sie beispielsweise 4.960 Leseanfragen mit einer Zeilengröße von 2,5 KB in Ihrer Cassandra-Tabelle ausführen, benötigen Sie 4.960 RCUs in Amazon Keyspaces. Wenn Sie derzeit 1.653 Schreibenforderungen pro Sekunde mit einer Zeilengröße von 2,5 KB in Ihrer Cassandra-Tabelle ausführen, benötigen Sie in Amazon Keyspaces 4.959 WCUs pro Sekunde. Dieses Beispiel wird in den folgenden Formeln ausgedrückt.

$$\begin{aligned} &4,960 \text{ read requests per second} * \text{ROUNDUP}(2.5\text{KB} / 4\text{KB bytes per unit}) \\ &= 4,960 \text{ read requests per second} * 1 \text{ RCU} \\ &= 4,960 \text{ RCUs} \end{aligned}$$

$$\begin{aligned} &1,653 \text{ write requests per second} * \text{ROUNDUP}(2.5\text{KB}/1\text{KB per unit}) \\ &= 1,653 \text{ requests per second} * 3 \text{ WCUs} \\ &= 4,959 \text{ WCUs} \end{aligned}$$

eventual consistency Durch die Verwendung können Sie bei jeder Leseanforderung bis zur Hälfte der Durchsatzkapazität sparen. Jeder eventuell konsistente Lesevorgang kann bis zu 8 KB verbrauchen. Sie können letztendlich konsistente Lesevorgänge berechnen, indem Sie die vorherige Berechnung mit 0,5 multiplizieren, wie in der folgenden Formel dargestellt.

$$\begin{aligned} &4,960 \text{ read requests per second} * \text{ROUNDUP}(2.5\text{KB} / 4\text{KB per unit}) * .5 \\ &= 2,480 \text{ read request per second} * 1 \text{ RCU} \\ &= 2,480 \text{ RCUs} \end{aligned}$$

5. Berechnen Sie die monatliche Preisschätzung für Amazon Keyspaces

Um die monatliche Abrechnung für die Tabelle auf der Grundlage des Durchsatzes der Lese-/Schreibkapazität zu schätzen, können Sie die Preise für den On-Demand-Modus und den Bereitstellungsmodus anhand verschiedener Formeln berechnen und die Optionen für Ihre Tabelle vergleichen.

Bereitgestellter Modus — Der Kapazitätsverbrauch für Lese- und Schreibvorgänge wird nach einem Stundensatz abgerechnet, der auf den Kapazitätseinheiten pro Sekunde basiert. Teilen Sie diese Rate zunächst durch 0,7, um die standardmäßige automatische Skalierungszielauslastung von 70% darzustellen. Dann multiplizieren Sie es mit 30 Kalendertagen, 24 Stunden pro Tag und den Preisen nach regionalen Tarifen. Diese Berechnung ist in den folgenden Formeln zusammengefasst.

```
(read capacity per second / .7) * 24 hours * 30 days * regional rate
(write capacity per second / .7) * 24 hours * 30 days * regional
rate
```

On-Demand-Modus — Lese- und Schreibkapazität werden pro Anforderung abgerechnet. Multiplizieren Sie zunächst die Anforderungsrate mit 30 Kalendertagen und 24 Stunden pro Tag. Teilen Sie dann durch eine Million Anforderungseinheiten. Schließlich multiplizieren Sie mit dem Regionalsatz. Diese Berechnung ist in den folgenden Formeln zusammengefasst.

```
((read capacity per second * 30 * 24 * 60 * 60) / 1 Million read request units) *
regional rate
((write capacity per second * 30 * 24 * 60 * 60) / 1 Million write
request units) * regional rate
```

Wählen Sie eine Migrationsstrategie

Im Allgemeinen können Sie bei der Migration von Apache Cassandra zu Amazon Keyspaces zwischen drei verschiedenen Migrationsstrategien wählen:

- **Offline** — Diese Migration beinhaltet das Kopieren eines Datensatzes von Cassandra nach Amazon Keyspaces mit einer Anwendungsmigrationsbereitstellung im blau/grünen Stil. Wenn Ihre Anwendung während der Migration einige Ausfallzeiten toleriert, kann diese Option den Migrationsprozess vereinfachen. Weitere Informationen zur Offline-Migration finden Sie unter [the section called “Offline-Migration”](#).
- **Online** — Dies ist eine Bereitstellung im kanarischen Stil, die in der Regel duale Schreibvorgänge beinhaltet, die direkt in die Anwendungslogik geschrieben werden. Bei Anwendungen, die während der Migration keine Ausfallzeiten erfordern, müssen Daten kopiert werden, während Lese- und Schreibvorgänge in Echtzeit von einer Datenquelle zur anderen umgeschaltet werden.

- Hybrid — Dieser Ansatz ermöglicht es, Änderungen nahezu in Echtzeit zu replizieren, aber die Anwendung ist für den Wechsel zwischen Lese- und Schreibvorgängen verantwortlich.

Nachdem Sie die verfügbaren Migrationsstrategien eingehender geprüft haben, können Sie die Optionen in einer Entscheidungsstruktur platzieren, um den Prozess unter Berücksichtigung Ihrer Anforderungen und verfügbaren Ressourcen zu vereinfachen.

Offline-Migration zu Amazon Keyspaces

Offline-Migrationen eignen sich, wenn Sie sich Ausfallzeiten für die Durchführung der Migration leisten können. In Unternehmen ist es üblich, Wartungsfenster für Patches, große Releases oder Ausfallzeiten für Hardware-Upgrades oder größere Upgrades vorzusehen. Die Offline-Migration kann dieses Fenster verwenden, um Daten zu kopieren und den Anwendungsdatenverkehr von Apache Cassandra zu Amazon Keyspaces umzuschalten. Die Offline-Migration reduziert die Anzahl der Änderungen an der Anwendung, da sie nicht gleichzeitig mit Cassandra und Amazon Keyspaces kommunizieren muss. Außerdem kann bei unterbrochenem Datenfluss der exakte Status kopiert werden, ohne dass Mutationen beibehalten werden.

In diesem Beispiel verwenden Sie Amazon Simple Storage Service (Amazon S3) als Staging-Bereich für Daten während der Offline-Migration, um Ausfallzeiten zu minimieren. Sie können die Daten, die Sie im Parquet-Format in Amazon S3 gespeichert haben, mithilfe des Spark-Cassandra-Connectors und automatisch in eine Amazon Keyspaces-Tabelle importieren. AWS Glue Der folgende Abschnitt gibt einen allgemeinen Überblick über den Prozess. Codebeispiele für diesen Prozess finden Sie auf [Github](#).

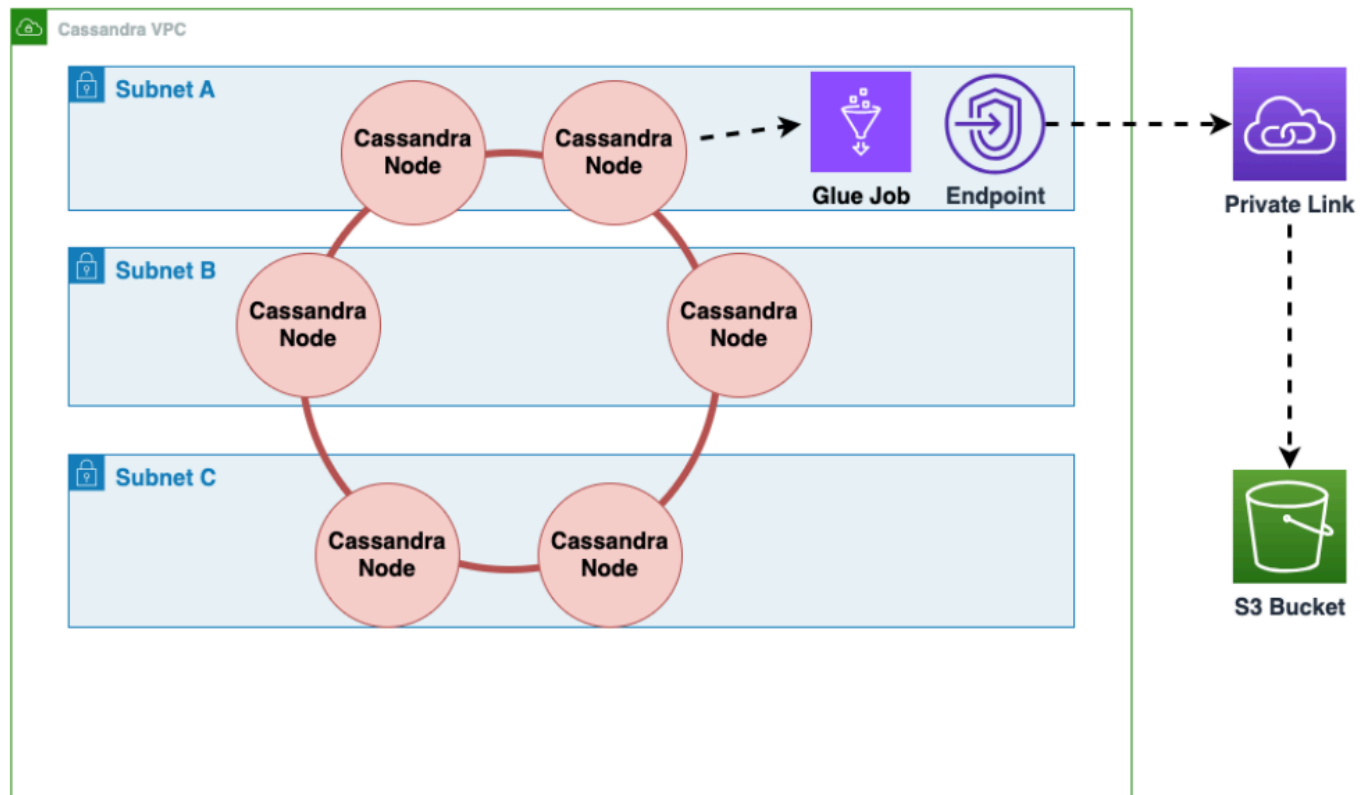
Der Offline-Migrationsprozess von Apache Cassandra zu Amazon Keyspaces mithilfe von Amazon S3 AWS Glue erfordert die folgenden AWS Glue Jobs.

1. Ein ETL-Job, der CQL-Daten extrahiert, transformiert und in einem Amazon S3 S3-Bucket speichert.
2. Ein zweiter Job, der die Daten aus dem Bucket in Amazon Keyspaces importiert.
3. Ein dritter Job zum Importieren inkrementeller Daten.

So führen Sie eine Offline-Migration von Cassandra, die auf Amazon EC2 in einer Amazon Virtual Private Cloud läuft, zu Amazon Keyspaces durch

1. Zuerst exportieren Sie AWS Glue Tabellendaten aus Cassandra im Parquet-Format und speichern sie in einem Amazon S3 S3-Bucket. Sie müssen einen AWS Glue Job mithilfe eines

AWS Glue Connectors zu einer VPC ausführen, auf der sich die Amazon EC2 EC2-Instance befindet, auf der Cassandra ausgeführt wird. Anschließend können Sie mit dem privaten Amazon S3 S3-Endpoint Daten im Amazon S3 S3-Bucket speichern. Das folgende Diagramm veranschaulicht diese Schritte.



2. Mischen Sie die Daten im Amazon S3 S3-Bucket, um die Datenrandomisierung zu verbessern. Gleichmäßig importierte Daten ermöglichen einen stärker verteilten Datenverkehr in der Zieltabelle. Dieser Schritt ist erforderlich, wenn Daten aus Cassandra mit großen Partitionen (Partitionen mit mehr als 1000 Zeilen) exportiert werden, um Tastenkombinationen beim Einfügen der Daten in Amazon Keyspaces zu vermeiden. Hotkey-Probleme treten `WriteThrottleEvents` in Amazon Keyspaces auf und führen zu einer längeren Ladezeit.



3. Verwenden Sie einen anderen AWS Glue Job, um Daten aus dem Amazon S3 S3-Bucket in Amazon Keyspaces zu importieren. Die gemischten Daten im Amazon S3 S3-Bucket werden im Parquet-Format gespeichert.



Tools für die Migration von Daten zu Amazon Keyspaces

Für die Migration von Daten zu Amazon Keyspaces stehen verschiedene Tools zur Verfügung

- Tools für die Migration
 - Erwägen Sie bei großen Migrationen die Verwendung eines ETL-Tool (Extrahieren, Transformieren und Laden). Sie können es verwenden AWS Glue , um Datentransformationsmigrationen schnell und effektiv durchzuführen.
 - Informationen zur Verwendung des Apache Cassandra Spark-Connectors zum Schreiben von Daten in Amazon Keyspaces finden Sie unter. [Integration mit Apache Spark](#)
 - Beginnen Sie schnell mit dem Laden von Daten in Amazon Keyspaces mithilfe des COPY FROM Befehls cqlsh. cqlsh ist in Apache Cassandra enthalten und eignet sich am besten zum Laden kleiner Datensätze oder Testdaten. [the section called "Laden von Daten mit cqlsh"](#)Eine Anleitung finden Sie step-by-step unter.
 - Sie können auch den DataStax Bulk Loader für Apache Cassandra verwenden, um Daten mit dem Befehl in Amazon Keyspaces zu laden. dsbulk [DSBulk bietet robustere Importfunktionen als cqlsh und ist im Repository verfügbar. GitHub](#) step-by-step Anweisungen finden Sie unter. [the section called "Daten mit DSBulk laden"](#)

Themen

- [Tutorial: Laden von Daten in Amazon Keyspaces mit cqlsh](#)
- [Tutorial: Daten mit DSBulk in Amazon Keyspaces laden](#)

Tutorial: Laden von Daten in Amazon Keyspaces mit cqlsh

Dieses step-by-step Tutorial führt Sie durch die Migration von Daten von Apache Cassandra zu Amazon Keyspaces mithilfe des `cqlsh COPY`-Befehls. In diesem Tutorial führen Sie folgende Aufgaben aus:

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen der CSV-Quelldatei und der Zieltabelle](#)
- [Schritt 2: Vorbereiten der Daten](#)
- [Schritt 3: Festlegen der Durchsatzkapazität für die Tabelle](#)
- [Schritt 4: Konfigurieren von cqlsh COPY FROM Einstellungen](#)
- [Schritt 5: Ausführen des cqlsh COPY FROM Befehls](#)
- [Fehlerbehebung](#)

Voraussetzungen

Sie müssen die folgenden Aufgaben ausführen, bevor Sie mit diesem Tutorial beginnen können.

1. Wenn Sie dies noch nicht getan haben, melden Sie sich für ein an, AWS-Konto indem Sie die Schritte unter [ausführenthe section called “Einrichten AWS Identity and Access Management”](#).
2. Erstellen Sie servicespezifische Anmeldeinformationen, indem Sie die Schritte unter [ausführenthe section called “Generieren Sie dienstspezifische Anmeldeinformationen mithilfe der Konsole”](#).
3. Richten Sie die Cassandra Query Language Shell (cqlsh)-Verbindung ein und bestätigen Sie, dass Sie eine Verbindung zu Amazon Keyspaces herstellen können, indem Sie die Schritte unter [ausführenthe section called “Verwenden von cqlsh”](#).

Schritt 1: Erstellen der CSV-Quelldatei und der Zieltabelle

Für dieses Tutorial verwenden wir eine CSV-Datei (durch Kommas getrennte Werte) mit dem Namen `keyspaces_sample_table.csv` als Quelldatei für die Datenmigration. Die bereitgestellte Beispieldatei enthält einige Datenzeilen für eine Tabelle mit dem Namen `book_awards`.

1. Erstellen Sie die Quelldatei. Sie können eine der folgenden Optionen wählen:

- Laden Sie die CSV-Beispieldatei (`keyspaces_sample_table.csv`) herunter, die in der folgenden Datei [samplmigration.zip](#) enthalten ist. Entpacken Sie das Archiv und notieren Sie sich den Pfad zu `keyspaces_sample_table.csv`.
- Um eine CSV-Datei mit Ihren eigenen Daten zu füllen, die in einer Apache-Cassandra-Datenbank gespeichert sind, können Sie die Quell-CSV-Datei mit der `-cqlshCOPY TO`Anweisung füllen, wie im folgenden Beispiel gezeigt.

```
cqlsh localhost 9042 -u "username" -p "password" --execute
"COPY mykeyspace.mytable TO 'keyspaces_sample_table.csv' WITH HEADER=true"
```

Stellen Sie sicher, dass die von Ihnen erstellte CSV-Datei die folgenden Anforderungen erfüllt:

- Die erste Zeile enthält die Spaltennamen.
 - Die Spaltennamen in der CSV-Quelldatei stimmen mit den Spaltennamen in der Zieltabelle überein.
 - Die Daten werden durch ein Komma getrennt.
 - Alle Datenwerte sind gültige Amazon Keyspaces-Datentypen. Siehe [the section called "Datentypen"](#).
2. Erstellen Sie den Ziel-Keyspace und die Tabelle in Amazon Keyspaces.
 - a. Stellen Sie mithilfe von eine Verbindung zu Amazon Keyspaces her `cqlsh` ersetzen Sie den Service-Endpunkt, den Benutzernamen und das Passwort im folgenden Beispiel durch Ihre eigenen Werte.

```
cqlsh cassandra.us-east-2.amazonaws.com 9142 -u "111122223333" -
p "wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY" --ssl
```

- b. Erstellen Sie einen neuen Keyspace mit dem Namen `catalog` wie im folgenden Beispiel gezeigt.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

- c. Wenn der neue Keyspace verfügbar ist, verwenden Sie den folgenden Code, um die Zieltabelle zu erstellen `book_awards`.

```
CREATE TABLE "catalog.book_awards" (
  year int,
```

```
award text,  
rank int,  
category text,  
book_title text,  
author text,  
publisher text,  
PRIMARY KEY ((year, award), category, rank)  
);
```

Wenn Apache Cassandra Ihre ursprüngliche Datenquelle ist, besteht eine einfache Möglichkeit, die Amazon Keyspaces-Zieltabelle mit übereinstimmenden Headern zu erstellen, darin, die CREATE TABLE Anweisung aus der Quelltable zu generieren, wie in der folgenden Anweisung gezeigt.

```
cqlsh localhost 9042 -u "username" -p "password" --execute "DESCRIBE  
TABLE mykeyspace.mytable;"
```

Erstellen Sie dann die Zieltabelle in Amazon Keyspaces mit den Spaltennamen und Datentypen, die der Beschreibung aus der Cassandra-Quelltable entsprechen.

Schritt 2: Vorbereiten der Daten

Die Vorbereitung der Quelldaten für eine effiziente Übertragung erfolgt in zwei Schritten. Zuerst randomisieren Sie die Daten. Im zweiten Schritt analysieren Sie die Daten, um die entsprechenden cqlsh Parameterwerte und erforderlichen Tabelleneinstellungen zu ermitteln.

Daten randomisieren

Der cqlsh COPY FROM Befehl liest und schreibt Daten in derselben Reihenfolge, in der sie in der CSV-Datei angezeigt werden. Wenn Sie den cqlsh COPY TO Befehl verwenden, um die Quelldatei zu erstellen, werden die Daten in schlüsselsortierter Reihenfolge in das CSV geschrieben. Intern partitioniert Amazon Keyspaces Daten mithilfe von Partitionsschlüsseln. Obwohl Amazon Keyspaces über eine integrierte Logik verfügt, um das Load Balancing von Anforderungen für denselben Partitionsschlüssel zu unterstützen, ist das Laden der Daten schneller und effizienter, wenn Sie die Reihenfolge randomisieren. Dies liegt daran, dass Sie das integrierte Load Balancing nutzen können, das auftritt, wenn Amazon Keyspaces auf verschiedene Partitionen schreibt.

Um die Schreibvorgänge gleichmäßig auf die Partitionen zu verteilen, müssen Sie die Daten in der Quelldatei randomisieren. Sie können eine Anwendung schreiben, um dies zu tun, oder ein

Open-Source-Tool wie [Shuf](#) verwenden. Shuf ist unter Linux-Distributionen, unter macOS (durch die Installation von coreutils in [homebrew](#)) und unter Windows (durch die Verwendung des Windows-Subsystems für Linux (WSL)) frei verfügbar. Ein zusätzlicher Schritt ist erforderlich, um zu verhindern, dass die Kopfzeile mit den Spaltennamen in diesem Schritt umgeleitet wird.

Geben Sie den folgenden Code ein, um die Quelldatei zu randomisieren und dabei den Header beizubehalten.

```
tail -n +2 keyspaces_sample_table.csv | shuf -o keyspace.table.csv && (head -1 keyspaces_sample_table.csv && cat keyspace.table.csv ) > keyspace.table.csv1 && mv keyspace.table.csv1 keyspace.table.csv
```

Shuf schreibt die Daten in eine neue CSV-Datei namens `keyspace.table.csv`. Sie können die `keyspaces_sample_table.csv` Datei jetzt löschen – Sie benötigen sie nicht mehr.

Analysieren der Daten

Ermitteln Sie die durchschnittliche und maximale Zeilengröße, indem Sie die Daten analysieren.

Sie tun dies aus folgenden Gründen:

- Die durchschnittliche Zeilengröße hilft bei der Schätzung der Gesamtmenge der zu übertragenden Daten.
- Sie benötigen die durchschnittliche Zeilengröße, um die für den Daten-Upload erforderliche Schreibkapazität bereitzustellen.
- Sie können sicherstellen, dass jede Zeile kleiner als 1 MB ist, was der maximalen Zeilengröße in Amazon Keyspaces entspricht.

Note

Dieses Kontingent bezieht sich auf die Zeilengröße, nicht auf die Partitionsgröße. Im Gegensatz zu Apache-Cassandra-Partitionen können Amazon-Keyspaces-Partitionen praktisch unbegrenzt groß sein. Partitionsschlüssel und Clustering-Spalten erfordern zusätzlichen Speicherplatz für Metadaten, den Sie der Rohgröße von Zeilen hinzufügen müssen. Weitere Informationen finden Sie unter [the section called “Berechnung der Zeilengröße”](#).

Der folgende Code verwendet [AWK](#), um eine CSV-Datei zu analysieren und die durchschnittliche und maximale Zeilengröße zu drucken.

```
awk -F, 'BEGIN {samp=10000;max=-1;}{if(NR>1){len=length($0);t+=len;avg=t/NR;max=(len>max ? len : max)}}NR==samp{exit}END{printf("[lines: %d, average: %d bytes, max: %d bytes]\n",NR,avg,max);}' keyspace.table.csv
```

Das Ausführen dieses Codes führt zu der folgenden Ausgabe.

```
using 10,000 samples:
{lines: 10000, avg: 123 bytes, max: 225 bytes}
```

Sie verwenden die durchschnittliche Zeilengröße im nächsten Schritt dieses Tutorials, um die Schreibkapazität für die Tabelle bereitzustellen.

Schritt 3: Festlegen der Durchsatzkapazität für die Tabelle

Dieses Tutorial zeigt Ihnen, wie Sie `cqlsh` so einstellen, dass Daten innerhalb eines festgelegten Zeitraums geladen werden. Da Sie wissen, wie viele Lese- und Schreibvorgänge Sie im Voraus ausführen, verwenden Sie den Modus bereitgestellter Kapazität. Nachdem Sie die Datenübertragung abgeschlossen haben, sollten Sie den Kapazitätsmodus der Tabelle so einstellen, dass er den Datenverkehrsmustern Ihrer Anwendung entspricht. Weitere Informationen zum Kapazitätsmanagement finden Sie unter [Serverloses Ressourcenmanagement](#).

Im Modus bereitgestellter Kapazität geben Sie im Voraus an, wie viel Lese- und Schreibkapazität Sie für Ihre Tabelle bereitstellen möchten. Die Schreibkapazität wird stündlich abgerechnet und in Schreibkapazitätseinheiten (WCUs) gemessen. Jede WCU verfügt über genügend Schreibkapazität, um das Schreiben von 1 KB Daten pro Sekunde zu unterstützen. Wenn Sie die Daten laden, muss die Schreibrate unter den maximalen WCUs (Parameter: `write_capacity_units`) liegen, die in der Zieltabelle festgelegt sind.

Standardmäßig können Sie bis zu 40.000 WCUs für eine Tabelle und 80.000 WCUs für alle Tabellen in Ihrem Konto bereitstellen. Wenn Sie zusätzliche Kapazität benötigen, können Sie eine Kontingenterhöhung in der [Service Quotas](#)-Konsole beantragen. Weitere Informationen zu Kontingenten finden Sie unter [Kontingente](#).

Berechnen der durchschnittlichen Anzahl von WCUs, die für eine Einfügung erforderlich sind

Das Einfügen von 1 KB Daten pro Sekunde erfordert 1 WCU. Wenn Ihre CSV-Datei 360.000 Zeilen enthält und Sie alle Daten in einer Stunde laden möchten, müssen Sie 100 Zeilen pro Sekunde

schreiben (360.000 Zeilen/60 Minuten/60 Sekunden = 100 Zeilen pro Sekunde). Wenn jede Zeile bis zu 1 KB Daten enthält, müssen Sie 100 WCUs für Ihre Tabelle bereitstellen, um 100 Zeilen pro Sekunde einzufügen. Wenn jede Zeile 1,5 KB Daten enthält, benötigen Sie zwei WCUs, um eine Zeile pro Sekunde einzufügen. Um 100 Zeilen pro Sekunde einzufügen, müssen Sie daher 200 WCUs bereitstellen.

Um zu bestimmen, wie viele WCUs Sie eine Zeile pro Sekunde einfügen müssen, dividieren Sie die durchschnittliche Zeilengröße in Bytes durch 1024 und auf die nächste ganze Zahl aufgerundet.

Wenn die durchschnittliche Zeilengröße beispielsweise 3000 Byte beträgt, benötigen Sie drei WCUs, um eine Zeile pro Sekunde einzufügen.

```
ROUNDUP(3000 / 1024) = ROUNDUP(2.93) = 3 WCUs
```

Berechnen der Datenladezeit und -kapazität

Da Sie nun die durchschnittliche Größe und Anzahl der Zeilen in Ihrer CSV-Datei kennen, können Sie berechnen, wie viele WCUs Sie in einer bestimmten Zeit laden müssen und wie lange es ungefähr dauert, alle Daten in Ihrer CSV-Datei mit unterschiedlichen WCU-Einstellungen zu laden.

Wenn jede Zeile in Ihrer Datei beispielsweise 1 KB groß ist und Sie 1 000 000 Zeilen in Ihrer CSV-Datei haben, müssen Sie für diese Stunde mindestens 278 WCUs für Ihre Tabelle bereitstellen, um die Daten innerhalb einer Stunde zu laden.

```
1,000,000 rows * 1 KBs = 1,000,000 KBs  
1,000,000 KBs / 3600 seconds = 277.8 KBs / second = 278 WCUs
```

Konfigurieren der Einstellungen für bereitgestellte Kapazität

Sie können die Schreibkapazitätseinstellungen einer Tabelle festlegen, wenn Sie die Tabelle erstellen, oder mit dem ALTER TABLE CQL-Befehl. Im Folgenden finden Sie die Syntax zum Ändern der bereitgestellten Kapazitätseinstellungen einer Tabelle mit der ALTER TABLE CQL-Anweisung.

```
ALTER TABLE mykeyspace.mytable WITH custom_properties={'capacity_mode':  
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 100,  
'write_capacity_units': 278}} ;
```

Die vollständige Sprachreferenz finden Sie unter [the section called “ALTER TABLE”](#).

Schritt 4: Konfigurieren von `cqlsh COPY FROM` Einstellungen

In diesem Abschnitt wird beschrieben, wie Sie die Parameterwerte für `cqlsh COPY FROM` festlegen. Der `cqlsh COPY FROM` Befehl liest die zuvor vorbereitete CSV-Datei und fügt die Daten mithilfe von CQL in Amazon Keyspaces ein. Der Befehl teilt die Zeilen auf und verteilt die INSERT Operationen auf eine Reihe von Workern. Jeder Worker stellt eine Verbindung mit Amazon Keyspaces her und sendet INSERT Anfragen entlang dieses Kanals.

Der `cqlsh COPY` Befehl verfügt nicht über eine interne Logik, um die Arbeit gleichmäßig auf die Worker zu verteilen. Sie können sie jedoch manuell konfigurieren, um sicherzustellen, dass die Arbeit gleichmäßig verteilt wird. Überprüfen Sie zunächst diese wichtigen `cqlsh`-Parameter:

- **DELIMITER** – Wenn Sie ein anderes Trennzeichen als ein Komma verwendet haben, können Sie diesen Parameter auf Komma setzen.
- **INGESTRATE** – Die Zielanzahl der Zeilen, die pro Sekunde zu verarbeiten `cqlsh COPY FROM` versucht. Wenn diese Einstellung nicht festgelegt ist, wird standardmäßig 100.000 verwendet.
- **NUMPROCESSES** – Die Anzahl der untergeordneten Worker-Prozesse, die `cqlsh` für `COPY FROM` Aufgaben erstellt. Das Maximum für diese Einstellung ist 16, der Standardwert ist `num_cores - 1`, wobei die Anzahl der Verarbeitungskerne auf dem Host `num_cores` ist, auf dem `cqlsh` ausgeführt wird.
- **MAXBATCHSIZE** – Die Batchgröße bestimmt die maximale Anzahl von Zeilen, die in einem einzigen Batch in die Zieltabelle eingefügt werden. Wenn diese Einstellung nicht festgelegt ist, verwendet `cqlsh` Batches mit 20 eingefügten Zeilen.
- **CHUNKSIZE** – Die Größe der Arbeitseinheit, die an den untergeordneten Worker übergibt. Standardmäßig ist er auf 5.000 festgelegt.
- **MAXATTEMPTS** – Die maximale Anzahl von Wiederholungsversuchen für einen fehlgeschlagenen Worker-Block. Nachdem der maximale Versuch erreicht ist, werden die fehlgeschlagenen Datensätze in eine neue CSV-Datei geschrieben, die Sie später erneut ausführen können, nachdem Sie den Fehler untersucht haben.

Legen Sie **INGESTRATE** basierend auf der Anzahl der WCUs fest, die Sie für die Zieltabelle bereitgestellt haben. Der **INGESTRATE** des `cqlsh COPY FROM` Befehls ist kein Limit, sondern ein Zieldurchschnitt. Dies bedeutet, dass es über die von Ihnen festgelegte Zahl hinaus gesteigert werden kann (und oft tut). Um Spitzen zu ermöglichen und sicherzustellen, dass genügend Kapazität vorhanden ist, um die Datenladeanforderungen zu verarbeiten, legen Sie **INGESTRATE** auf 90 % der Schreibkapazität der Tabelle fest.

```
INGESTRATE = WCUs * .90
```

Legen Sie als Nächstes den NUMPROCESSES Parameter auf gleich eins kleiner als die Anzahl der Kerne auf Ihrem System fest. Um die Anzahl der Kerne Ihres Systems zu ermitteln, können Sie den folgenden Code ausführen.

```
python -c "import multiprocessing; print(multiprocessing.cpu_count())"
```

Für dieses Tutorial verwenden wir den folgenden Wert.

```
NUMPROCESSES = 4
```

Jeder Prozess erstellt einen Worker und jeder Worker stellt eine Verbindung zu Amazon Keyspaces her. Amazon Keyspaces kann bis zu 3 000 CQL-Anforderungen pro Sekunde für jede Verbindung unterstützen. Das bedeutet, dass Sie sicherstellen müssen, dass jeder Worker weniger als 3 000 Anfragen pro Sekunde verarbeitet.

Wie bei werden INGESTRATE die Auftragnehmer häufig über die von Ihnen festgelegte Zahl hinaus gesteigert und sind nicht durch Taktsekunden begrenzt. Um Spitzen zu berücksichtigen, legen Sie daher Ihre cqlsh-Parameter so fest, dass jeder Worker 2 500 Anfragen pro Sekunde verarbeitet. Verwenden Sie die folgende Richtlinie, um die Menge der an einen Worker verteilten Arbeit zu berechnen.

- Dividieren INGESTRATE Sie durch NUMPROCESSES.
- Wenn $INGESTRATE / NUMPROCESSES > 2\,500$, senken Sie die , INGESTRATE um diese Formel zu „true“ zu machen.

```
INGESTRATE / NUMPROCESSES <= 2,500
```

Bevor Sie die Einstellungen zur Optimierung des Uploads unserer Beispieldaten konfigurieren, überprüfen wir die cqlsh Standardeinstellungen und sehen, wie sich die Verwendung auf den Daten-Upload-Prozess auswirkt. Da die cqlsh COPY FROM verwendet, CHUNKSIZE um Arbeitsblöcke (INSERT-Anweisungen) für die Verteilung an Worker zu erstellen, wird die Arbeit nicht automatisch gleichmäßig verteilt. Je nach INGESTRATE Einstellung können einige Mitarbeiter inaktiv sein.

Um die Arbeit gleichmäßig auf die Worker zu verteilen und jeden Worker mit der optimalen Rate von 2 500 Anforderungen pro Sekunde zu halten, müssen Sie CHUNKSIZE, und

festlegen `MAXBATCHSIZE`, `INGESTRATE` indem Sie die Eingabeparameter ändern. Um die Auslastung des Netzwerkverkehrs während des Datenladevorgangs zu optimieren, wählen Sie einen Wert für `MAXBATCHSIZE`, der nahe dem Maximalwert von 30 liegt. Durch die Änderung `CHUNKSIZE` auf 100 und `MAXBATCHSIZE` 25 werden die 10 000 Zeilen gleichmäßig auf die vier Worker verteilt ($10\,000 / 2\,500 = 4$).

Das folgende Codebeispiel veranschaulicht dies.

```
INGESTRATE = 10,000
NUMPROCESSES = 4
CHUNKSIZE = 100
MAXBATCHSIZE. = 25
Work Distribution:
Connection 1 / Worker 1 : 2,500 Requests per second
Connection 2 / Worker 2 : 2,500 Requests per second
Connection 3 / Worker 3 : 2,500 Requests per second
Connection 4 / Worker 4 : 2,500 Requests per second
```

Verwenden Sie zum Zusammenfassen die folgenden Formeln, wenn Sie `cqlsh COPY FROM` Parameter festlegen:

- $INGESTRATE = write_capacity_units * 0,90$
- $NUMPROCESSES = num_cores - 1$ (Standard)
- $INGESTRATE / NUMPROCESSES = 2\,500$ (Dies muss eine wahre Anweisung sein.)
- $MAXBATCHSIZE = 30$ (standardmäßig 20. Amazon Keyspaces akzeptiert Batches bis zu 30.)
- $CHUNKSIZE = (AUFNAHME / NUMPROCESSES) / MAXBATCHSIZE$

Nachdem Sie `NUMPROCESSES`, `INGESTRATE` und berechnet haben `CHUNKSIZE`, können Sie Ihre Daten laden.

Schritt 5: Ausführen des `cqlsh COPY FROM` Befehls

Führen Sie die folgenden Schritte aus, um den `cqlsh COPY FROM` Befehl auszuführen.

1. Stellen Sie mithilfe von `cqlsh` eine Verbindung zu Amazon Keyspaces her.
2. Wählen Sie Ihren Keyspace mit dem folgenden Code aus.

```
USE catalog;
```

3. Legen Sie die Schreibkonsistenz auf festLOCAL_QUORUM. Um die Datenbeständigkeit zu gewährleisten, lässt Amazon Keyspaces keine anderen Einstellungen für die Schreibkonsistenz zu. Sehen Sie sich den folgenden Code an.

```
CONSISTENCY LOCAL_QUORUM;
```

4. Bereiten Sie Ihre cqlsh COPY FROM Syntax mit dem folgenden Codebeispiel vor.

```
COPY book_awards FROM './keyspace.table.csv' WITH HEADER=true  
AND INGESTRATE=calculated ingestrate  
AND NUMPROCESSES=calculated numprocess  
AND MAXBATCHSIZE=20  
AND CHUNKSIZE=calculated chunksize;
```

5. Führen Sie die im vorherigen Schritt vorbereitete Anweisung aus. cqlsh gibt alle von Ihnen konfigurierten Einstellungen zurück.
 - a. Stellen Sie sicher, dass die Einstellungen mit Ihrer Eingabe übereinstimmen. Sehen Sie sich das folgende -Beispiel an.

```
Reading options from the command line: {'chunksize': '120', 'header': 'true',  
  'ingestrate': '36000', 'numprocesses': '15', 'maxbatchsize': '20'}  
Using 15 child processes
```

- b. Überprüfen Sie die Anzahl der übertragenen Zeilen und die aktuelle durchschnittliche Rate, wie im folgenden Beispiel gezeigt.

```
Processed: 57834 rows; Rate: 6561 rows/s; Avg. rate: 31751 rows/s
```

- c. Wenn cqlsh mit dem Hochladen der Daten fertig ist, überprüfen Sie die Zusammenfassung der Datenladestatistiken (die Anzahl der gelesenen, Laufzeit- und übersprungenen Dateien), wie im folgenden Beispiel gezeigt.

```
15556824 rows imported from 1 files in 8 minutes and 8.321 seconds (0 skipped).
```

In diesem letzten Schritt des Tutorials haben Sie die Daten in Amazon Keyspaces hochgeladen.

Important

Nachdem Sie Ihre Daten übertragen haben, passen Sie die Kapazitätsmoduseinstellungen Ihrer Zieltabelle an die regulären Datenverkehrsmuster Ihrer Anwendung an. Es fallen Gebühren zum Stundensatz für Ihre bereitgestellte Kapazität an, bis Sie sie ändern.

Fehlerbehebung

Überprüfen Sie nach Abschluss des Daten-Uploads, ob Zeilen übersprungen wurden. Navigieren Sie dazu zum Quellverzeichnis der CSV-Quelldatei und suchen Sie nach einer Datei mit dem folgenden Namen.

```
import_yourcsvfilename.err.timestamp.csv
```

cqlsh schreibt alle übersprungenen Datenzeilen in eine Datei mit diesem Namen. Wenn die Datei in Ihrem Quellverzeichnis vorhanden ist und Daten enthält, wurden diese Zeilen nicht in Amazon Keyspaces hochgeladen. Um diese Zeilen erneut zu versuchen, überprüfen Sie zunächst auf Fehler, die während des Uploads aufgetreten sind, und passen Sie die Daten entsprechend an. Um diese Zeilen erneut zu versuchen, können Sie den Prozess erneut ausführen.

Häufige Fehler

Die häufigsten Gründe, warum Zeilen nicht geladen werden, sind Kapazitätsfehler und Analysefehler.

Ungültige Anforderungsfehler beim Hochladen von Daten in Amazon Keyspaces

Im folgenden Beispiel enthält die Quelltable eine Zählerspalte, die zu protokollierten Batch-Aufrufen aus dem COPY Befehl cqlsh führt. Protokollierte Batch-Aufrufe werden von Amazon Keyspaces nicht unterstützt.

```
Failed to import 10 rows: InvalidRequest - Error from server: code=2200 [Invalid query]
message="Only UNLOGGED Batches are supported at this time.", will retry later,
attempt 22 of 25
```

Um diesen Fehler zu beheben, verwenden Sie DSBulk, um die Daten zu migrieren. Weitere Informationen finden Sie unter [the section called "Daten mit DSBulk laden"](#).

Parserfehler beim Hochladen von Daten in Amazon Keyspaces

Das folgende Beispiel zeigt eine übersprungene Zeile aufgrund eines `ParseError`.

```
Failed to import 1 rows: ParseError - Invalid ... -
```

Um diesen Fehler zu beheben, müssen Sie sicherstellen, dass die zu importierenden Daten mit dem Tabellenschema in Amazon Keyspaces übereinstimmen. Überprüfen Sie die Importdatei auf Analysefehler. Sie können versuchen, eine einzelne Datenzeile mit einer `-INSERT`Anweisung zu verwenden, um den Fehler zu isolieren.

Kapazitätsfehler beim Hochladen von Daten in Amazon Keyspaces

```
Failed to import 1 rows: WriteTimeout - Error from server: code=1100 [Coordinator node
  timed out waiting for replica nodes' responses]
  message="Operation timed out - received only 0 responses." info={'received_responses':
  0, 'required_responses': 2, 'write_type': 'SIMPLE', 'consistency':
  'LOCAL_QUORUM'}, will retry later, attempt 1 of 100
```

Amazon Keyspaces verwendet die `WriteTimeout` Ausnahmen `ReadTimeout` und `WriteTimeout`, um anzugeben, wann eine Schreibanforderung aufgrund unzureichender Durchsatzkapazität fehlschlägt. Um bei der Diagnose von Ausnahmen mit unzureichender Kapazität zu helfen, veröffentlicht Amazon Keyspaces `-WriteThrottleEvents` und `-ReadThrottledEvents` Metriken in Amazon CloudWatch. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

cqlsh-Fehler beim Hochladen von Daten in Amazon Keyspaces

Um die Behebung von `cqlsh`-Fehlern zu unterstützen, führen Sie den fehlgeschlagenen Befehl mit dem `---debugFlag` erneut aus.

Wenn Sie eine inkompatible Version von `cqlsh` verwenden, wird der folgende Fehler angezeigt.

```
AttributeError: 'NoneType' object has no attribute 'is_up'
Failed to import 3 rows: AttributeError - 'NoneType' object has no attribute 'is_up',
  given up after 1 attempts
```

Bestätigen Sie, dass die richtige Version von `cqlsh` installiert ist, indem Sie den folgenden Befehl ausführen.

```
cqlsh --version
```


Zur Ausgabe sollte etwas wie das folgende angezeigt werden.

```
cqlsh 5.0.1
```

Wenn Sie Windows verwenden, ersetzen Sie alle Instances von `cqlsh` durch `cqlsh.bat`. Um beispielsweise die Version von `cqlsh` unter Windows zu überprüfen, führen Sie den folgenden Befehl aus.

```
cqlsh.bat --version
```

Die Verbindung zu Amazon Keyspaces schlägt fehl, nachdem der `cqlsh`-Client drei aufeinanderfolgende Fehler eines beliebigen Typs vom Server erhalten hat. Der `cqlsh`-Client schlägt mit der folgenden Meldung fehl.

```
Failed to import 1 rows: NoHostAvailable - , will retry later, attempt 3 of 100
```

Um diesen Fehler zu beheben, müssen Sie sicherstellen, dass die zu importierenden Daten mit dem Tabellenschema in Amazon Keyspaces übereinstimmen. Überprüfen Sie die Importdatei auf Analysefehler. Sie können versuchen, eine einzelne Datenzeile zu verwenden, indem Sie eine `INSERT`-Anweisung verwenden, um den Fehler zu isolieren.

Der Client versucht automatisch, die Verbindung wiederherzustellen.

Tutorial: Daten mit DSBulk in Amazon Keyspaces laden

Dieses step-by-step Tutorial führt Sie durch die Migration von Daten von Apache Cassandra zu Amazon Keyspaces mithilfe des DataStax Bulk Loaders (DSBulk), der auf [verfügbar](#) ist. [GitHub](#) In diesem Tutorial führen Sie die folgenden Schritte aus:

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie die Quell-CSV-Datei und die Zieltabelle](#)
- [Schritt 2: Bereiten Sie die Daten vor](#)
- [Schritt 3: Stellen Sie die Durchsatzkapazität für die Tabelle ein](#)
- [Schritt 4: DSBulk Einstellungen konfigurieren](#)
- [Schritt 5: Führen Sie den Befehl DSBulk load aus](#)

Voraussetzungen

Sie müssen die folgenden Aufgaben erledigen, bevor Sie mit diesem Tutorial beginnen können.

1. Falls Sie dies noch nicht getan haben, registrieren Sie sich für ein AWS Konto, indem Sie den Schritten unter folgen [the section called “Einrichten AWS Identity and Access Management”](#).
2. Erstellen Sie Anmeldeinformationen, indem Sie den Schritten unter folgen [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).
3. Erstellen Sie eine JKS-Trust-Store-Datei.
 - a. Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secyserver.net/repository/sf-class2-root.crt -O
```

Note

Sie können das digitale Amazon-Zertifikat auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen, und Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

- b. Konvertieren Sie das digitale Starfield-Zertifikat in eine TrustStore-Datei.

```
openssl x509 -outform der -in sf-class2-root.crt -out temp_file.der  
keytool -import -alias cassandra -keystore cassandra_truststore.jks -file  
temp_file.der
```

In diesem Schritt müssen Sie ein Passwort für den Keystore erstellen und diesem Zertifikat vertrauen. Der interaktive Befehl sieht so aus.

```
Enter keystore password:  
Re-enter new password:  
Owner: OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,  
Inc.", C=US  
Issuer: OU=Starfield Class 2 Certification Authority, O="Starfield  
Technologies, Inc.", C=US  
Serial number: 0
```

```

Valid from: Tue Jun 29 17:39:16 UTC 2004 until: Thu Jun 29 17:39:16 UTC 2034
Certificate fingerprints:
  MD5: 32:4A:4B:BB:C8:63:69:9B:BE:74:9A:C6:DD:1D:46:24
  SHA1: AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A
  SHA256:
  14:65:FA:20:53:97:B8:76:FA:A6:F0:A9:95:8E:55:90:E4:0F:CC:7F:AA:4F:B7:C2:C8:67:75:21:FB
Signature algorithm name: SHA1withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3
Extensions:
#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....
]
[OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
Inc.", C=US]
SerialNumber: [ 00]
]
#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
CA:true
PathLen:2147483647
]
#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....
]
]
Trust this certificate? [no]: y

```

4. Richten Sie die Cassandra Query Language Shell (cqlsh) -Verbindung ein und bestätigen Sie, dass Sie eine Verbindung zu Amazon Keyspaces herstellen können, indem Sie die Schritte unter befolgen. [the section called "Verwenden von cqlsh"](#)
5. Laden Sie DSBulk herunter und installieren Sie es.
 - a. Um DSBulk herunterzuladen, können Sie den folgenden Code verwenden.

```
curl -OL https://downloads.datastax.com/dsbulk/dsbulk-1.8.0.tar.gz
```

- b. Entpacken Sie dann die TAR-Datei und fügen Sie DSBulk zu Ihrer hinzu, PATH wie im folgenden Beispiel gezeigt.

```
tar -zxvf dsbulk-1.8.0.tar.gz
# add the DSBulk directory to the path
export PATH=$PATH:./dsbulk-1.8.0/bin
```

- c. Erstellen Sie eine `application.conf` Datei zum Speichern von Einstellungen, die von DSBulk verwendet werden sollen. Sie können das folgende Beispiel unter `speichern. ./dsbulk_keyspaces.conf` `localhost` Ersetzen Sie es durch den Kontaktpunkt Ihres lokalen Cassandra-Clusters, wenn Sie sich nicht auf dem lokalen Knoten befinden, z. B. den DNS-Namen oder die IP-Adresse. Notieren Sie sich den Dateinamen und den Pfad, da Sie dies später im `dsbulk load` Befehl angeben müssen.

```
datastax-java-driver {
  basic.contact-points = [ "localhost" ]
  advanced.auth-provider {
    class = software.aws.mcs.auth.SigV4AuthProvider
    aws-region = us-east-1
  }
}
```

- d. Um die SigV4-Unterstützung zu aktivieren, laden Sie die schattierte `jar` Datei von [herunter GitHub](#) und platzieren Sie sie im `lib` Ordner DSBulk, wie im folgenden Beispiel gezeigt.

```
curl -O -L https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin/releases/download/4.0.6-shaded-v2/aws-sigv4-auth-cassandra-java-driver-plugin-4.0.6-shaded.jar
```

Schritt 1: Erstellen Sie die Quell-CSV-Datei und die Zieltabelle

Für dieses Tutorial verwenden wir eine Datei mit kommagetrennten Werten (CSV) mit dem Namen `keyspaces_sample_table.csv` als Quelldatei für die Datenmigration. Die mitgelieferte Beispieldatei enthält einige Datenzeilen für eine Tabelle mit dem Namen `book_awards`

1. Erstellen Sie die Quelldatei. Sie können eine der folgenden Optionen wählen:
 - Laden Sie die CSV-Beispieldatei (`keyspaces_sample_table.csv`) herunter, die in der folgenden Archivdatei [samplmigration.zip](#) enthalten ist. Entpacken Sie das Archiv und notieren Sie sich den Pfad zu `keyspaces_sample_table.csv`.

- Um eine CSV-Datei mit Ihren eigenen Daten zu füllen, die in einer Apache Cassandra-Datenbank gespeichert sind, können Sie die CSV-Quelldatei `dsbulk unload` wie im folgenden Beispiel gezeigt auffüllen.

```
dsbulk unload -k mykeyspace -t mytable -f ./my_application.conf  
> keyspaces_sample_table.csv
```

Stellen Sie sicher, dass die von Ihnen erstellte CSV-Datei die folgenden Anforderungen erfüllt:

- Die erste Zeile enthält die Spaltennamen.
- Die Spaltennamen in der CSV-Quelldatei stimmen mit den Spaltennamen in der Zieltabelle überein.
- Die Daten sind durch ein Komma getrennt.
- Alle Datenwerte sind gültige Amazon Keyspaces-Datentypen. Siehe [the section called “Datentypen”](#).

2. Erstellen Sie den Zielschlüsselraum und die Zieltabelle in Amazon Keyspaces.

- a. Connect zu Amazon Keyspaces `hercqlsh`, indem Sie den Service-Endpunkt, den Benutzernamen und das Passwort im folgenden Beispiel durch Ihre eigenen Werte ersetzen.

```
cqlsh cassandra.us-east-2.amazonaws.com 9142 -u "111122223333" -  
p "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY" --ssl
```

- b. Erstellen Sie einen neuen Schlüsselraum mit dem Namen, `catalog` wie im folgenden Beispiel gezeigt.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

- c. Wenn der neue Schlüsselraum den Status verfügbar hat, verwenden Sie den folgenden Code, um die Zieltabelle zu erstellen. `book_awards` Weitere Informationen zur asynchronen Ressourcenerstellung und zur Überprüfung, ob eine Ressource verfügbar ist, finden Sie unter [the section called “Erstellen von Keyspaces”](#)

```
CREATE TABLE catalog.book_awards (  
  year int,  
  award text,  
  rank int,
```

```
category text,  
book_title text,  
author text,  
publisher text,  
PRIMARY KEY ((year, award), category, rank)  
);
```

Wenn Apache Cassandra Ihre ursprüngliche Datenquelle ist, besteht eine einfache Möglichkeit, die Amazon Keyspaces-Zieltabelle mit passenden Headern zu erstellen, darin, die CREATE TABLE Anweisung aus der Quelltable zu generieren, wie in der folgenden Anweisung gezeigt.

```
cqlsh localhost 9042 -u "username" -p "password" --execute "DESCRIBE  
TABLE mykeyspace.mytable;"
```

Erstellen Sie dann die Zieltabelle in Amazon Keyspaces mit den Spaltennamen und Datentypen, die der Beschreibung aus der Cassandra-Quelltable entsprechen.

Schritt 2: Bereiten Sie die Daten vor

Die Vorbereitung der Quelldaten für eine effiziente Übertragung erfolgt in zwei Schritten. Zunächst randomisieren Sie die Daten. Im zweiten Schritt analysieren Sie die Daten, um die geeigneten dsbulk Parameterwerte und erforderlichen Tabelleneinstellungen zu ermitteln.

Randomisieren Sie die Daten

Der dsbulk Befehl liest und schreibt Daten in derselben Reihenfolge, in der sie in der CSV-Datei erscheinen. Wenn Sie den dsbulk Befehl verwenden, um die Quelldatei zu erstellen, werden die Daten in schlüsselsortierter Reihenfolge in die CSV-Datei geschrieben. Intern partitioniert Amazon Keyspaces Daten mithilfe von Partitionsschlüsseln. Amazon Keyspaces verfügt zwar über eine integrierte Logik, die beim Lastenausgleich von Anfragen für denselben Partitionsschlüssel hilft, das Laden der Daten ist jedoch schneller und effizienter, wenn Sie die Reihenfolge nach dem Zufallsprinzip festlegen. Dies liegt daran, dass Sie den integrierten Lastenausgleich nutzen können, der auftritt, wenn Amazon Keyspaces auf verschiedene Partitionen schreibt.

Um die Schreibvorgänge gleichmäßig auf die Partitionen zu verteilen, müssen Sie die Daten in der Quelldatei randomisieren. [Sie können dafür eine Anwendung schreiben oder ein Open-Source-Tool wie Shuf verwenden](#). Shuf ist auf Linux-Distributionen, auf macOS (durch Installation von Coreutils in [Homebrew](#)) und unter Windows (mithilfe des Windows Subsystems for Linux (WSL)) kostenlos

verfügbar. Ein zusätzlicher Schritt ist erforderlich, um zu verhindern, dass die Kopfzeile mit den Spaltennamen in diesem Schritt gemischt wird.

Um die Quelldatei nach dem Zufallsprinzip zu sortieren und gleichzeitig die Kopfzeile beizubehalten, geben Sie den folgenden Code ein.

```
tail -n +2 keyspaces_sample_table.csv | shuf -o keyspace.table.csv && (head
-1 keyspaces_sample_table.csv && cat keyspace.table.csv ) > keyspace.table.csv1 &&
mv keyspace.table.csv1 keyspace.table.csv
```

Shuf schreibt die Daten in eine neue CSV-Datei mit dem Namen `keyspace.table.csv`. Sie können die `keyspaces_sample_table.csv` Datei jetzt löschen — Sie benötigen sie nicht mehr.

Analysieren Sie die Daten

Ermitteln Sie die durchschnittliche und maximale Zeilengröße, indem Sie die Daten analysieren.

Sie tun dies aus den folgenden Gründen:

- Die durchschnittliche Zeilengröße hilft bei der Schätzung der Gesamtmenge der zu übertragenden Daten.
- Sie benötigen die durchschnittliche Zeilengröße, um die für den Datenupload benötigte Schreibkapazität bereitzustellen.
- Sie können sicherstellen, dass jede Zeile weniger als 1 MB groß ist. Dies ist die maximale Zeilengröße in Amazon Keyspaces.

Note

Dieses Kontingent bezieht sich auf die Zeilengröße, nicht auf die Partitionsgröße. Im Gegensatz zu Apache Cassandra-Partitionen können Amazon Keyspaces-Partitionen praktisch unbegrenzt groß sein. Partitionsschlüssel und Clusterspalten benötigen zusätzlichen Speicherplatz für Metadaten, den Sie zur Rohgröße der Zeilen hinzufügen müssen. Weitere Informationen finden Sie unter [the section called “Berechnung der Zeilengröße”](#).

Der folgende Code verwendet [AWK](#), um eine CSV-Datei zu analysieren und die durchschnittliche und maximale Zeilengröße zu drucken.

```
awk -F, 'BEGIN {samp=10000;max=-1;}{if(NR>1){len=length($0);t+=len;avg=t/
NR;max=(len>max ? len : max)}}NR==samp{exit}END{printf("{lines: %d, average: %d bytes,
max: %d bytes}\n",NR,avg,max);}' keyspace.table.csv
```

Die Ausführung dieses Codes führt zu der folgenden Ausgabe.

```
using 10,000 samples:
{lines: 10000, avg: 123 bytes, max: 225 bytes}
```

Stellen Sie sicher, dass Ihre maximale Zeilengröße 1 MB nicht überschreitet. Ist dies der Fall, müssen Sie die Zeile aufteilen oder die Daten komprimieren, um die Zeilengröße unter 1 MB zu bringen. Im nächsten Schritt dieses Tutorials verwenden Sie die durchschnittliche Zeilengröße, um die Schreibkapazität für die Tabelle bereitzustellen.

Schritt 3: Stellen Sie die Durchsatzkapazität für die Tabelle ein

Dieses Tutorial zeigt Ihnen, wie Sie DSBulk so einstellen, dass Daten innerhalb eines festgelegten Zeitbereichs geladen werden. Da Sie im Voraus wissen, wie viele Lese- und Schreibvorgänge Sie durchführen, sollten Sie den Modus für bereitgestellte Kapazität verwenden. Nachdem Sie die Datenübertragung abgeschlossen haben, sollten Sie den Kapazitätsmodus der Tabelle so einstellen, dass er den Datenverkehrsmustern Ihrer Anwendung entspricht. Weitere Informationen zur Kapazitätsverwaltung finden Sie unter [Serverloses Ressourcenmanagement](#).

Im Modus „Bereitgestellte Kapazität“ geben Sie im Voraus an, wie viel Lese- und Schreibkapazität Sie für Ihre Tabelle bereitstellen möchten. Die Schreibkapazität wird stündlich abgerechnet und in Schreibkapazitätseinheiten (WCUs) gemessen. Jede WCU bietet ausreichend Schreibkapazität, um das Schreiben von 1 KB Daten pro Sekunde zu unterstützen. Wenn Sie die Daten laden, muss die Schreibrate unter den maximalen WCUs (Parameter: `write_capacity_units`) liegen, die in der Zieltabelle festgelegt sind.

Standardmäßig können Sie bis zu 40.000 WCUs für eine Tabelle und 80.000 WCUs für alle Tabellen in Ihrem Konto bereitstellen. Wenn Sie zusätzliche Kapazität benötigen, können Sie in der [Service Quotas-Konsole eine Erhöhung des Kontingents](#) beantragen. Weitere Informationen zu Kontingenten finden Sie unter [Kontingente](#).

Berechnen Sie die durchschnittliche Anzahl der WCUs, die für eine Einfügung erforderlich sind

Für das Einfügen von 1 KB Daten pro Sekunde ist 1 WCU erforderlich. Wenn Ihre CSV-Datei 360.000 Zeilen hat und Sie alle Daten in einer Stunde laden möchten, müssen Sie 100 Zeilen pro Sekunde

schreiben (360.000 Zeilen/60 Minuten/ 60 Sekunden = 100 Zeilen pro Sekunde). Wenn jede Zeile bis zu 1 KB Daten enthält, müssen Sie 100 WCUs für Ihre Tabelle bereitstellen, um 100 Zeilen pro Sekunde einzufügen. Wenn jede Zeile 1,5 KB Daten enthält, benötigen Sie zwei WCUs, um eine Zeile pro Sekunde einzufügen. Um 100 Zeilen pro Sekunde einzufügen, müssen Sie daher 200 WCUs bereitstellen.

Um zu ermitteln, wie viele WCUs Sie für das Einfügen einer Zeile pro Sekunde benötigen, teilen Sie die durchschnittliche Zeilengröße in Byte durch 1024 und runden Sie auf die nächste ganze Zahl auf.

Wenn die durchschnittliche Zeilengröße beispielsweise 3000 Byte beträgt, benötigen Sie drei WCUs, um eine Zeile pro Sekunde einzufügen.

```
ROUNDUP(3000 / 1024) = ROUNDUP(2.93) = 3 WCUs
```

Berechnen Sie die Ladezeit und Kapazität der Daten

Da Sie nun die durchschnittliche Größe und Anzahl der Zeilen in Ihrer CSV-Datei kennen, können Sie berechnen, wie viele WCUs Sie benötigen, um die Daten in einem bestimmten Zeitraum zu laden, und wie lange es ungefähr dauert, alle Daten in Ihrer CSV-Datei mithilfe verschiedener WCU-Einstellungen zu laden.

Wenn beispielsweise jede Zeile in Ihrer Datei 1 KB groß ist und Sie 1.000.000 Zeilen in Ihrer CSV-Datei haben, müssen Sie für diese Stunde mindestens 278 WCUs für Ihre Tabelle bereitstellen, um die Daten in einer Stunde zu laden.

```
1,000,000 rows * 1 KBs = 1,000,000 KBs  
1,000,000 KBs / 3600 seconds = 277.8 KBs / second = 278 WCUs
```

Konfigurieren Sie die Einstellungen für die bereitgestellte Kapazität

Sie können die Schreibkapazitätseinstellungen einer Tabelle festlegen, wenn Sie die Tabelle erstellen oder den ALTER TABLE Befehl verwenden. Im Folgenden finden Sie die Syntax für das Ändern der bereitgestellten Kapazitätseinstellungen einer Tabelle mit dem ALTER TABLE Befehl.

```
ALTER TABLE catalog.book_awards WITH custom_properties={'capacity_mode':  
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 100, 'write_capacity_units':  
278}} ;
```

Die vollständige Sprachreferenz finden Sie unter [the section called “CREATE TABLE”](#) und [the section called “ALTER TABLE”](#)

Schritt 4: **DSBulk** Einstellungen konfigurieren

In diesem Abschnitt werden die Schritte beschrieben, die erforderlich sind, um DSBulk für das Hochladen von Daten auf Amazon Keyspaces zu konfigurieren. Sie konfigurieren DSBulk mithilfe einer Konfigurationsdatei. Sie geben die Konfigurationsdatei direkt von der Befehlszeile aus an.

1. Erstellen Sie eine DSBulk-Konfigurationsdatei für die Migration zu Amazon Keyspaces. In diesem Beispiel verwenden wir den Dateinamen `dsbulk_keyspaces.conf`. Geben Sie die folgenden Einstellungen in der DSBulk-Konfigurationsdatei an.
 - a. *PlainTextAuthProvider*— Erstellen Sie den Authentifizierungsanbieter mit der `PlainTextAuthProvider` Klasse. `ServiceUserName` und `ServicePassword` sollte mit dem Benutzernamen und dem Passwort übereinstimmen, die Sie bei der Generierung der dienstspezifischen Anmeldeinformationen erhalten haben, indem Sie die Schritte unter [the section called "Anmeldeinformationen erstellen"](#) ausführen.
 - b. *local-datacenter*— Setzen Sie den Wert für `local-datacenter` auf den AWS-Region, zu dem Sie eine Verbindung herstellen. Wenn die Anwendung beispielsweise eine Verbindung herstellt `cassandra.us-east-2.amazonaws.com`, stellen Sie das lokale Rechenzentrum auf `us-east-2`. Alle verfügbaren AWS-Regionen Informationen finden Sie unter [the section called "Service-Endpunkte"](#). Um Replikate zu vermeiden, legen Sie den Wert `slow-replica-avoidance` auf `false` fest.
 - c. *SSLConnectionFactory*— Um SSL/TLS zu konfigurieren, initialisieren Sie das, `SSLConnectionFactory` indem Sie der Konfigurationsdatei einen Abschnitt mit einer einzigen Zeile hinzufügen, in der die Klasse mit angegeben wird. `class = DefaultSslConnectionFactory` Geben Sie den Pfad `cassandra_truststore.jks` und das Passwort an, die Sie zuvor erstellt haben.
 - d. *consistency*— Stellen Sie die Konsistenzstufe auf ein `LOCAL QUORUM`. Andere Schreibkonsistenzstufen werden nicht unterstützt. Weitere Informationen finden Sie unter [the section called "Unterstützte Cassandra-Konsistenzstufen"](#).
 - e. Die Anzahl der Verbindungen pro Pool ist im Java-Treiber konfigurierbar. Stellen Sie in diesem Beispiel `advanced.connection.pool.local.size` den Wert 3 ein.

Im Folgenden finden Sie die vollständige Beispielkonfigurationsdatei.

```
datastax-java-driver {
  basic.contact-points = [ "cassandra.us-east-2.amazonaws.com:9142" ]
  advanced.auth-provider {
```

```
class = PlainTextAuthProvider
username = "ServiceUserName"
password = "ServicePassword"
}

basic.load-balancing-policy {
  local-datacenter = "us-east-2"
  slow-replica-avoidance = false
}

basic.request {
  consistency = LOCAL_QUORUM
  default-idempotence = true
}

advanced.ssl-engine-factory {
  class = DefaultSslEngineFactory
  truststore-path = "./cassandra_truststore.jks"
  truststore-password = "my_password"
  hostname-validation = false
}

advanced.connection.pool.local.size = 3
}
```

2. Überprüfen Sie die Parameter für den load Befehl DSBulk.

- a. *executor.maxPerSecond*— Die maximale Anzahl von Zeilen, die der Ladebefehl pro Sekunde gleichzeitig zu verarbeiten versucht. Wenn sie nicht gesetzt ist, ist diese Einstellung mit -1 deaktiviert.

Wird auf der *executor.maxPerSecond* Grundlage der Anzahl der WCUs festgelegt, die Sie für die Zieltabelle bereitgestellt haben. Der *executor.maxPerSecond* Wert des load Befehls ist kein Limit, sondern ein Zieldurchschnitt. Das bedeutet, dass er die von Ihnen festgelegte Zahl überschreiten kann (und tut dies häufig auch). Um Bursts zu berücksichtigen und sicherzustellen, dass genügend Kapazität zur Bearbeitung der Datenladeanforderungen vorhanden ist, sollten Sie einen Wert von 90% der Schreibkapazität der Tabelle festlegen *executor.maxPerSecond*.

```
executor.maxPerSecond = WCUs * .90
```

In diesem Tutorial haben wir den Wert *executor.maxPerSecond* auf 5 gesetzt.

Note

Wenn Sie DSBulk 1.6.0 oder höher verwenden, können Sie stattdessen `verwendendsbulk.engine.maxConcurrentQueries`.

- b. Konfigurieren Sie diese zusätzlichen Parameter für den `load` DSBulk-Befehl.
- `batch-mode`— Dieser Parameter weist das System an, Operationen nach Partitionsschlüsseln zu gruppieren. Wir empfehlen, den Batch-Modus zu deaktivieren, da dies zu Hotkey-Szenarien und Ursachen führen kann `WriteThrottleEvents`.
 - `driver.advanced.retry-policy-max-retries`— Dies bestimmt, wie oft eine fehlgeschlagene Abfrage wiederholt werden muss. Wenn diese Option nicht gesetzt ist, ist die Standardeinstellung 10. Sie können diesen Wert nach Bedarf anpassen.
 - `driver.basic.request.timeout`— Die Zeit in Minuten, in der das System auf die Rückgabe einer Abfrage wartet. Wenn diese Option nicht gesetzt ist, ist die Standardeinstellung „5 Minuten“. Sie können diesen Wert nach Bedarf anpassen.

Schritt 5: Führen Sie den Befehl DSBulk **load** aus

Im letzten Schritt dieses Tutorials laden Sie die Daten in Amazon Keyspaces hoch.

Führen Sie die folgenden Schritte aus, um den `load` DSBulk-Befehl auszuführen.

1. Führen Sie den folgenden Code aus, um die Daten aus Ihrer CSV-Datei in Ihre Amazon Keyspaces-Tabelle hochzuladen. Achten Sie darauf, den Pfad zur Anwendungskonfigurationsdatei zu aktualisieren, die Sie zuvor erstellt haben.

```
dsbulk load -f ./dsbulk_keyspaces.conf --connector.csv.url keyspace.table.csv  
-header true --batch.mode DISABLED --executor.maxPerSecond 5 --  
driver.basic.request.timeout "5 minutes" --driver.advanced.retry-policy.max-  
retries 10 -k catalog -t book_awards
```

2. Die Ausgabe enthält den Speicherort einer Protokolldatei, in der erfolgreiche und erfolglose Vorgänge aufgeführt sind. Die Datei wird im folgenden Verzeichnis gespeichert.

```
Operation directory: /home/user_name/logs/UNLOAD_20210308-202317-801911
```

- Die Einträge in der Protokolldatei werden Metriken enthalten, wie im folgenden Beispiel. Stellen Sie sicher, dass die Anzahl der Zeilen mit der Anzahl der Zeilen in Ihrer CSV-Datei übereinstimmt.

```
total | failed | rows/s | p50ms | p99ms | p999ms
200 | 0 | 200 | 21.63 | 21.89 | 21.89
```

Important

Nachdem Sie Ihre Daten übertragen haben, passen Sie die Einstellungen für den Kapazitätsmodus Ihrer Zieltabelle an die regulären Datenverkehrsmuster Ihrer Anwendung an. Es fallen Gebühren zum Stundensatz für Ihre bereitgestellte Kapazität an, bis Sie diese ändern. Weitere Informationen finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).

Codebeispiele für Amazon Keyspaces mit AWS SDKs

Die folgenden Codebeispiele zeigen, wie Amazon Keyspaces mit einem AWS Software Development Kit (SDK) verwendet wird.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Servicefunktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien und serviceübergreifenden Beispiele sehen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie eine bestimmte Aufgabe ausführen können, indem Sie mehrere Funktionen innerhalb desselben Services aufrufen.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Erste Schritte

Hallo Amazon Keyspaces

Die folgenden Codebeispiele zeigen, wie Sie mit Amazon Keyspaces beginnen können.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
namespace KeyspacesActions;

public class HelloKeyspaces
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
```

```

        // Set up dependency injection for Amazon Keyspaces (for Apache
        Cassandra).
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
                    LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
                    LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonKeyspaces>()
                    .AddTransient<KeyspacesWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<HelloKeyspaces>();

        var keyspacesClient =
            host.Services.GetRequiredService<IAmazonKeyspaces>();
        var keyspacesWrapper = new KeyspacesWrapper(keyspacesClient);

        Console.WriteLine("Hello, Amazon Keyspaces! Let's list your keyspaces:");
        await keyspacesWrapper.ListKeyspaces();
    }
}

```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.keyspaces.KeyspacesClient;
import software.amazon.awssdk.services.keyspaces.model.KeyspaceSummary;
import software.amazon.awssdk.services.keyspaces.model.KeyspacesException;
import software.amazon.awssdk.services.keyspaces.model.ListKeyspacesRequest;
import software.amazon.awssdk.services.keyspaces.model.ListKeyspacesResponse;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class HelloKeyspaces {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        KeyspacesClient keyClient = KeyspacesClient.builder()
            .region(region)
            .build();

        listKeyspaces(keyClient);
    }

    public static void listKeyspaces(KeyspacesClient keyClient) {
        try {
            ListKeyspacesRequest keyspacesRequest =
                ListKeyspacesRequest.builder()
                    .maxResults(10)
                    .build();

            ListKeyspacesResponse response =
                keyClient.listKeyspaces(keyspacesRequest);
            List<KeyspaceSummary> keyspaces = response.keyspaces();
            for (KeyspaceSummary keyspace : keyspaces) {
                System.out.println("The name of the keyspace is " +
                    keyspace.keyspaceName());
            }
        } catch (KeyspacesException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```



```
    }  
  }  
}
```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/**  
Before running this Kotlin code example, set up your development environment,  
including your credentials.  
  
For more information, see the following documentation topic:  
  
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html  
*/  
  
suspend fun main() {  
    listKeyspaces()  
}  

```

```
}  
}
```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
import boto3  
  
def hello_keyspaces(keyspaces_client):  
    """  
    Use the AWS SDK for Python (Boto3) to create an Amazon Keyspaces (for Apache  
    Cassandra)  
    client and list the keyspaces in your account.  
    This example uses the default settings specified in your shared credentials  
    and config files.  
  
    :param keyspaces_client: A Boto3 Amazon Keyspaces Client object. This object  
    wraps  
                                the low-level Amazon Keyspaces service API.  
    """  
    print("Hello, Amazon Keyspaces! Let's list some of your keyspaces:\n")  
    for ks in keyspaces_client.list_keyspaces(maxResults=5).get("keyspaces", []):  
        print(ks["keyspaceName"])  
        print(f"\t{ks['resourceArn']}")  
  
if __name__ == "__main__":  
    hello_keyspaces(boto3.client("keyspaces"))
```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in AWS SDK for Python (Boto3) API Reference.

Codebeispiele

- [Aktionen für Amazon Keyspaces mithilfe von AWS SDKs](#)
 - [Verwendung CreateKeyspace mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateTable mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteKeyspace mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteTable mit einem AWS SDK oder CLI](#)
 - [Verwendung GetKeyspace mit einem AWS SDK oder CLI](#)
 - [Verwendung GetTable mit einem AWS SDK oder CLI](#)
 - [Verwendung ListKeyspaces mit einem AWS SDK oder CLI](#)
 - [Verwendung ListTables mit einem AWS SDK oder CLI](#)
 - [Verwendung RestoreTable mit einem AWS SDK oder CLI](#)
 - [Verwendung UpdateTable mit einem AWS SDK oder CLI](#)
- [Szenarien für Amazon Keyspaces mit AWS SDKs](#)
 - [Erste Schritte mit Amazon Keyspaces-Keyspaces und -Tabellen mithilfe eines SDK AWS](#)

Aktionen für Amazon Keyspaces mithilfe von AWS SDKs

Die folgenden Codebeispiele zeigen, wie einzelne Amazon Keyspaces-Aktionen mit AWS SDKs durchgeführt werden. Diese Auszüge rufen die Amazon Keyspaces-API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [Amazon Keyspaces \(für Apache Cassandra\) API-Referenz](#).

Beispiele

- [Verwendung CreateKeyspace mit einem AWS SDK oder CLI](#)
- [Verwendung CreateTable mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteKeyspace mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteTable mit einem AWS SDK oder CLI](#)

- [Verwendung GetKeyspace mit einem AWS SDK oder CLI](#)
- [Verwendung GetTable mit einem AWS SDK oder CLI](#)
- [Verwendung ListKeyspaces mit einem AWS SDK oder CLI](#)
- [Verwendung ListTables mit einem AWS SDK oder CLI](#)
- [Verwendung RestoreTable mit einem AWS SDK oder CLI](#)
- [Verwendung UpdateTable mit einem AWS SDK oder CLI](#)

Verwendung **CreateKeyspace** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `CreateKeyspace`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Create a new keyspace.
/// </summary>
/// <param name="keyspaceName">The name for the new keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the new keyspace.</returns>
public async Task<string> CreateKeyspace(string keyspaceName)
{
    var response =
        await _amazonKeyspaces.CreateKeyspaceAsync(
            new CreateKeyspaceRequest { KeyspaceName = keyspaceName });
    return response.ResourceArn;
}
```

- Einzelheiten zur API finden Sie [CreateKeyspace](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void createKeySpace(KeyspacesClient keyClient, String
keyspaceName) {
    try {
        CreateKeyspaceRequest keyspaceRequest =
CreateKeyspaceRequest.builder()
            .keyspaceName(keyspaceName)
            .build();

        CreateKeyspaceResponse response =
keyClient.createKeyspace(keyspaceRequest);
        System.out.println("The ARN of the KeySpace is " +
response.resourceArn());

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Einzelheiten zur API finden Sie [CreateKeyspace](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
suspend fun createKeySpace(keyspaceNameVal: String) {
    val keyspaceRequest =
        CreateKeyspaceRequest {
            keyspaceName = keyspaceNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        val response = keyClient.createKeyspace(keyspaceRequest)
        println("The ARN of the KeySpace is ${response.resourceArn}")
    }
}
```

- Einzelheiten zur API finden Sie [CreateKeyspace](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""
```

```
def __init__(self, keyspaces_client):
    """
    :param keyspaces_client: A Boto3 Amazon Keyspaces client.
    """
    self.keyspaces_client = keyspaces_client
    self.ks_name = None
    self.ks_arn = None
    self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def create_keyspace(self, name):
        """
        Creates a keyspace.

        :param name: The name to give the keyspace.
        :return: The Amazon Resource Name (ARN) of the new keyspace.
        """
        try:
            response = self.keyspaces_client.create_keyspace(keyspaceName=name)
            self.ks_name = name
            self.ks_arn = response["resourceArn"]
        except ClientError as err:
            logger.error(
                "Couldn't create %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return self.ks_arn
```

- Einzelheiten zur API finden Sie [CreateKeyspace](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **CreateTable** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `CreateTable`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Create a new Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace where the table will be
created.</param>
/// <param name="schema">The schema for the new table.</param>
/// <param name="tableName">The name of the new table.</param>
/// <returns>The Amazon Resource Name (ARN) of the new table.</returns>
public async Task<string> CreateTable(string keyspaceName, SchemaDefinition
schema, string tableName)
{
    var request = new CreateTableRequest
    {
        KeyspaceName = keyspaceName,
        SchemaDefinition = schema,
        TableName = tableName,
        PointInTimeRecovery = new PointInTimeRecovery { Status =
PointInTimeRecoveryStatus.ENABLED }
    }
}
```



```
};

var response = await _amazonKeyspaces.CreateTableAsync(request);
return response.ResourceArn;
}
```

- Einzelheiten zur API finden Sie [CreateTable](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void createTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
    try {
        // Set the columns.
        ColumnDefinition defTitle = ColumnDefinition.builder()
            .name("title")
            .type("text")
            .build();

        ColumnDefinition defYear = ColumnDefinition.builder()
            .name("year")
            .type("int")
            .build();

        ColumnDefinition defReleaseDate = ColumnDefinition.builder()
            .name("release_date")
            .type("timestamp")
            .build();

        ColumnDefinition defPlot = ColumnDefinition.builder()
            .name("plot")
            .type("text")
```

```
        .build();

        List<ColumnDefinition> collist = new ArrayList<>();
        collist.add(defTitle);
        collist.add(defYear);
        collist.add(defReleaseDate);
        collist.add(defPlot);

        // Set the keys.
        PartitionKey yearKey = PartitionKey.builder()
            .name("year")
            .build();

        PartitionKey titleKey = PartitionKey.builder()
            .name("title")
            .build();

        List<PartitionKey> keyList = new ArrayList<>();
        keyList.add(yearKey);
        keyList.add(titleKey);

        SchemaDefinition schemaDefinition = SchemaDefinition.builder()
            .partitionKeys(keyList)
            .allColumns(collist)
            .build();

        PointInTimeRecovery timeRecovery = PointInTimeRecovery.builder()
            .status(PointInTimeRecoveryStatus.ENABLED)
            .build();

        CreateTableRequest tableRequest = CreateTableRequest.builder()
            .keyspaceName(keySpace)
            .tableName(tableName)
            .schemaDefinition(schemaDefinition)
            .pointInTimeRecovery(timeRecovery)
            .build();

        CreateTableResponse response = keyClient.createTable(tableRequest);
        System.out.println("The table ARN is " + response.resourceArn());

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

```
}
```

- Einzelheiten zur API finden Sie [CreateTable](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
suspend fun createTable(
    keySpaceVal: String?,
    tableNameVal: String?,
) {
    // Set the columns.
    val defTitle =
        ColumnDefinition {
            name = "title"
            type = "text"
        }

    val defYear =
        ColumnDefinition {
            name = "year"
            type = "int"
        }

    val defReleaseDate =
        ColumnDefinition {
            name = "release_date"
            type = "timestamp"
        }

    val defPlot =
        ColumnDefinition {
            name = "plot"
            type = "text"
        }
}
```

```
    }

    val collist = ArrayList<ColumnDefinition>()
    collist.add(defTitle)
    collist.add(defYear)
    collist.add(defReleaseDate)
    collist.add(defPlot)

    // Set the keys.
    val yearKey =
        PartitionKey {
            name = "year"
        }

    val titleKey =
        PartitionKey {
            name = "title"
        }

    val keyList = ArrayList<PartitionKey>()
    keyList.add(yearKey)
    keyList.add(titleKey)

    val schemaDefinition0b =
        SchemaDefinition {
            partitionKeys = keyList
            allColumns = collist
        }

    val timeRecovery =
        PointInTimeRecovery {
            status = PointInTimeRecoveryStatus.Enabled
        }

    val tableRequest =
        CreateTableRequest {
            keyspaceName = keySpaceVal
            tableName = tableNameVal
            schemaDefinition = schemaDefinition0b
            pointInTimeRecovery = timeRecovery
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        val response = keyClient.createTable(tableRequest)
    }
```

```

        println("The table ARN is ${response.resourceArn}")
    }
}

```

- Einzelheiten zur API finden Sie [CreateTable](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def create_table(self, table_name):
        """
        Creates a table in the keyspace.
        The table is created with a schema for storing movie data
        and has point-in-time recovery enabled.

```

```
:param table_name: The name to give the table.
:return: The ARN of the new table.
"""
try:
    response = self.keyspaces_client.create_table(
        keyspaceName=self.ks_name,
        tableName=table_name,
        schemaDefinition={
            "allColumns": [
                {"name": "title", "type": "text"},
                {"name": "year", "type": "int"},
                {"name": "release_date", "type": "timestamp"},
                {"name": "plot", "type": "text"},
            ],
            "partitionKeys": [{"name": "year"}, {"name": "title"}],
        },
        pointInTimeRecovery={"status": "ENABLED"},
    )
except ClientError as err:
    logger.error(
        "Couldn't create table %s. Here's why: %s: %s",
        table_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
else:
    return response["resourceArn"]
```

- Einzelheiten zur API finden Sie [CreateTable](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeleteKeyspace** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DeleteKeyspace`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Delete an existing keyspace.
/// </summary>
/// <param name="keyspaceName"></param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteKeyspace(string keyspaceName)
{
    var response = await _amazonKeyspaces.DeleteKeyspaceAsync(
        new DeleteKeyspaceRequest { KeyspaceName = keyspaceName });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Einzelheiten zur API finden Sie [DeleteKeyspace](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void deleteKeyspace(KeyspacesClient keyClient, String
keyspaceName) {
    try {
        DeleteKeyspaceRequest deleteKeyspaceRequest =
DeleteKeyspaceRequest.builder()
            .keyspaceName(keyspaceName)
            .build();

        keyClient.deleteKeyspace(deleteKeyspaceRequest);

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Einzelheiten zur API finden Sie [DeleteKeyspace](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
suspend fun deleteKeyspace(keyspaceNameVal: String?) {
    val deleteKeyspaceRequest =
        DeleteKeyspaceRequest {
            keyspaceName = keyspaceNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient.deleteKeyspace(deleteKeyspaceRequest)
    }
}
```


- Einzelheiten zur API finden Sie [DeleteKeyspace](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def delete_keyspace(self):
        """
        Deletes the keyspace.
        """
        try:
            self.keyspaces_client.delete_keyspace(keyspaceName=self.ks_name)
            self.ks_name = None
        except ClientError as err:
            logger.error(
```

```
        "Couldn't delete keyspace %s. Here's why: %s: %s",
        self.ks_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
```

- Einzelheiten zur API finden Sie [DeleteKeyspace](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeleteTable** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `DeleteTable`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Delete an Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table to delete.</param>
```

```
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteTable(string keyspaceName, string tableName)
{
    var response = await _amazonKeyspaces.DeleteTableAsync(
        new DeleteTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
    return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Einzelheiten zur API finden Sie [DeleteTable](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void deleteTable(KeyspacesClient keyClient, String
keyspaceName, String tableName) {
    try {
        DeleteTableRequest tableRequest = DeleteTableRequest.builder()
            .keyspaceName(keyspaceName)
            .tableName(tableName)
            .build();

        keyClient.deleteTable(tableRequest);

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Einzelheiten zur API finden Sie [DeleteTable](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
suspend fun deleteTable(
    keyspaceNameVal: String?,
    tableNameVal: String?,
) {
    val tableRequest =
        DeleteTableRequest {
            keyspaceName = keyspaceNameVal
            tableName = tableNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient.deleteTable(tableRequest)
    }
}
```

- Einzelheiten zur API finden Sie [DeleteTable](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:
```

```
"""Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
actions."""

def __init__(self, keyspaces_client):
    """
    :param keyspaces_client: A Boto3 Amazon Keyspaces client.
    """
    self.keyspaces_client = keyspaces_client
    self.ks_name = None
    self.ks_arn = None
    self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def delete_table(self):
        """
        Deletes the table from the keyspace.
        """
        try:
            self.keyspaces_client.delete_table(
                keyspaceName=self.ks_name, tableName=self.table_name
            )
            self.table_name = None
        except ClientError as err:
            logger.error(
                "Couldn't delete table %s. Here's why: %s: %s",
                self.table_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Einzelheiten zur API finden Sie [DeleteTable](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetKeyspace** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `GetKeyspace`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Get data about a keyspace.
/// </summary>
/// <param name="keyspaceName">The name of the keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the keyspace.</returns>
public async Task<string> GetKeyspace(string keyspaceName)
{
    var response = await _amazonKeyspaces.GetKeyspaceAsync(
        new GetKeyspaceRequest { KeyspaceName = keyspaceName });
    return response.ResourceArn;
}
```

- Einzelheiten zur API finden Sie [GetKeyspace](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void checkKeyspaceExistence(KeyspacesClient keyClient, String
keyspaceName) {
    try {
        GetKeyspaceRequest keyspaceRequest = GetKeyspaceRequest.builder()
            .keyspaceName(keyspaceName)
            .build();

        GetKeyspaceResponse response =
keyClient.getKeyspace(keyspaceRequest);
        String name = response.keyspaceName();
        System.out.println("The " + name + " KeySpace is ready");

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Einzelheiten zur API finden Sie [GetKeyspace](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
suspend fun checkKeyspaceExistence(keyspaceNameVal: String?) {
    val keyspaceRequest =
        GetKeyspaceRequest {
            keyspaceName = keyspaceNameVal
        }
    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        val response: GetKeyspaceResponse =
            keyClient.getKeyspace(keyspaceRequest)
        val name = response.keyspaceName
        println("The $name KeySpace is ready")
    }
}
```

- Einzelheiten zur API finden Sie [GetKeyspace](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
```



```
keyspaces_client = boto3.client("keyspaces")
return cls(keyspaces_client)

def exists_keyspace(self, name):
    """
    Checks whether a keyspace exists.

    :param name: The name of the keyspace to look up.
    :return: True when the keyspace exists. Otherwise, False.
    """
    try:
        response = self.keyspaces_client.get_keyspace(keyspaceName=name)
        self.ks_name = response["keyspaceName"]
        self.ks_arn = response["resourceArn"]
        exists = True
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.info("Keyspace %s does not exist.", name)
            exists = False
        else:
            logger.error(
                "Couldn't verify %s exists. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
    return exists
```

- Einzelheiten zur API finden Sie [GetKeyspace](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetTable** mit einem AWS SDK oder CLI


Die folgenden Codebeispiele zeigen, wie es verwendet wird `GetTable`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Get information about an Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the Amazon Keyspaces table.</param>
/// <returns>The response containing data about the table.</returns>
public async Task<GetTableResponse> GetTable(string keyspaceName, string
tableName)
{
    var response = await _amazonKeyspaces.GetTableAsync(
        new GetTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
    return response;
}
```

- Einzelheiten zur API finden Sie [GetTable](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void checkTable(KeyspacesClient keyClient, String keyspaceName,
String tableName)
    throws InterruptedException {
    try {
        boolean tableStatus = false;
        String status;
        GetTableResponse response = null;
        GetTableRequest tableRequest = GetTableRequest.builder()
            .keyspaceName(keyspaceName)
            .tableName(tableName)
            .build();

        while (!tableStatus) {
            response = keyClient.getTable(tableRequest);
            status = response.statusAsString();
            System.out.println(". The table status is " + status);

            if (status.compareTo("ACTIVE") == 0) {
                tableStatus = true;
            }
            Thread.sleep(500);
        }

        List<ColumnDefinition> cols =
response.schemaDefinition().allColumns();
        for (ColumnDefinition def : cols) {
            System.out.println("The column name is " + def.name());
            System.out.println("The column type is " + def.type());
        }

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
    }
}
```

```

        System.exit(1);
    }
}

```

- Einzelheiten zur API finden Sie [GetTable](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

suspend fun checkTable(
    keyspaceNameVal: String?,
    tableNameVal: String?,
) {
    var tableStatus = false
    var status: String
    var response: GetTableResponse? = null

    val tableRequest =
        GetTableRequest {
            keyspaceName = keyspaceNameVal
            tableName = tableNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        while (!tableStatus) {
            response = keyClient.getTable(tableRequest)
            status = response!!.status.toString()
            println(". The table status is $status")
            if (status.compareTo("ACTIVE") == 0) {
                tableStatus = true
            }
            delay(500)
        }
        val cols: List<ColumnDefinition>? =
            response!!.schemaDefinition?.allColumns
    }
}

```

```

        if (cols != null) {
            for (def in cols) {
                println("The column name is ${def.name}")
                println("The column type is ${def.type}")
            }
        }
    }
}

```

- Einzelheiten zur API finden Sie [GetTable](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def get_table(self, table_name):

```

```
"""
Gets data about a table in the keyspace.

:param table_name: The name of the table to look up.
:return: Data about the table.
"""
try:
    response = self.keyspaces_client.get_table(
        keyspaceName=self.ks_name, tableName=table_name
    )
    self.table_name = table_name
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.info("Table %s does not exist.", table_name)
        self.table_name = None
        response = None
    else:
        logger.error(
            "Couldn't verify %s exists. Here's why: %s: %s",
            table_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
return response
```

- Einzelheiten zur API finden Sie [GetTable](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **ListKeyspaces** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListKeyspaces`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Lists all keyspaces for the account.
/// </summary>
/// <returns>Async task.</returns>
public async Task ListKeyspaces()
{
    var paginator = _amazonKeyspaces.Paginators.ListKeyspaces(new
ListKeyspacesRequest());

    Console.WriteLine("{0, -30}\t{1}", "Keyspace name", "Keyspace ARN");
    Console.WriteLine(new string('-', Console.WindowWidth));
    await foreach (var keyspace in paginator.Keyspaces)
    {
        Console.WriteLine($"{keyspace.KeyspaceName, -30}\t{keyspace.ResourceArn}");
    }
}
```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

public static void listKeyspacesPaginator(KeyspacesClient keyClient) {
    try {
        ListKeyspacesRequest keyspacesRequest =
ListKeyspacesRequest.builder()
                        .maxResults(10)
                        .build();

        ListKeyspacesIterable listRes =
keyClient.listKeyspacesPaginator(keyspacesRequest);
        listRes.stream()
                .flatMap(r -> r.keyspaces().stream())
                .forEach(content -> System.out.println(" Name: " +
content.keyspaceName()));

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

suspend fun listKeyspacesPaginator() {
    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient
            .listKeyspacesPaginated(ListKeyspacesRequest {})
            .transform { it.keyspaces?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name: ${obj.keyspaceName}")
            }
    }
}

```



```
}  
}
```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:  
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table  
    actions."""  
  
    def __init__(self, keyspaces_client):  
        """  
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.  
        """  
        self.keyspaces_client = keyspaces_client  
        self.ks_name = None  
        self.ks_arn = None  
        self.table_name = None  
  
    @classmethod  
    def from_client(cls):  
        keyspaces_client = boto3.client("keyspaces")  
        return cls(keyspaces_client)  
  
    def list_keyspaces(self, limit):  
        """  
        Lists the keyspaces in your account.  
  
        :param limit: The maximum number of keyspaces to list.  
        """  
        try:
```

```
ks_paginator = self.keyspaces_client.get_paginator("list_keyspaces")
for page in ks_paginator.paginate(PaginationConfig={"MaxItems":
limit}):
    for ks in page["keyspaces"]:
        print(ks["keyspaceName"])
        print(f"\t{ks['resourceArn']}")
except ClientError as err:
    logger.error(
        "Couldn't list keyspaces. Here's why: %s: %s",
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
```

- Einzelheiten zur API finden Sie [ListKeyspaces](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **ListTables** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `ListTables`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

/// <summary>
/// Lists the Amazon Keyspaces tables in a keyspace.
/// </summary>
/// <param name="keyspaceName">The name of the keyspace.</param>
/// <returns>A list of TableSummary objects.</returns>
public async Task<List<TableSummary>> ListTables(string keyspaceName)
{
    var response = await _amazonKeyspaces.ListTablesAsync(new
ListTablesRequest { KeyspaceName = keyspaceName });
    response.Tables.ForEach(table =>
    {
        Console.WriteLine($"{table.KeyspaceName}\t{table.TableName}\t{table.ResourceArn}");
    });

    return response.Tables;
}

```

- Einzelheiten zur API finden Sie [ListTables](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

public static void listTables(KeyspacesClient keyClient, String keyspaceName)
{
    try {
        ListTablesRequest tablesRequest = ListTablesRequest.builder()
            .keyspaceName(keyspaceName)
            .build();

        ListTablesIterable listRes =
keyClient.listTablesPaginator(tablesRequest);

```

```

        listRes.stream()
            .flatMap(r -> r.tables().stream())
            .forEach(content -> System.out.println(" ARN: " +
content.resourceArn() +
                " Table name: " + content.tableName()));

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

- Einzelheiten zur API finden Sie [ListTables](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

suspend fun listTables(keyspaceNameVal: String?) {
    val tablesRequest =
        ListTablesRequest {
            keyspaceName = keyspaceNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient
            .listTablesPaginated(tablesRequest)
            .transform { it.tables?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println(" ARN: ${obj.resourceArn} Table name: ${obj.tableName}")
            }
    }
}

```

- Einzelheiten zur API finden Sie [ListTables](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. GitHub Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def list_tables(self):
        """
        Lists the tables in the keyspace.
        """
        try:
            table_paginator = self.keyspaces_client.get_paginator("list_tables")
            for page in table_paginator.paginate(keyspaceName=self.ks_name):
                for table in page["tables"]:
                    print(table["tableName"])
                    print(f"\t{table['resourceArn']}")
        except ClientError as err:
```

```
logger.error(  
    "Couldn't list tables in keyspace %s. Here's why: %s: %s",  
    self.ks_name,  
    err.response["Error"]["Code"],  
    err.response["Error"]["Message"],  
)  
raise
```

- Einzelheiten zur API finden Sie [ListTables](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **RestoreTable** mit einem AWS SDK oder CLI

Die folgenden Codebeispiele zeigen, wie es verwendet wird `RestoreTable`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>  
/// Restores the specified table to the specified point in time.  
/// </summary>  
/// <param name="keyspaceName">The keyspace containing the table.</param>
```

```
/// <param name="tableName">The name of the table to restore.</param>
/// <param name="timestamp">The time to which the table will be restored.</
param>
/// <returns>The Amazon Resource Name (ARN) of the restored table.</returns>
public async Task<string> RestoreTable(string keyspaceName, string tableName,
string restoredTableName, DateTime timestamp)
{
    var request = new RestoreTableRequest
    {
        RestoreTimestamp = timestamp,
        SourceKeyspaceName = keyspaceName,
        SourceTableName = tableName,
        TargetKeyspaceName = keyspaceName,
        TargetTableName = restoredTableName
    };

    var response = await _amazonKeyspaces.RestoreTableAsync(request);
    return response.RestoredTableARN;
}
```

- Einzelheiten zur API finden Sie [RestoreTable](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void restoreTable(KeyspacesClient keyClient, String
keyspaceName, ZonedDateTime utc) {
    try {
        Instant myTime = utc.toInstant();
        RestoreTableRequest restoreTableRequest =
RestoreTableRequest.builder()
            .restoreTimestamp(myTime)
            .sourceTableName("Movie")
```

```

        .targetKeyspaceName(keyspaceName)
        .targetTableName("MovieRestore")
        .sourceKeyspaceName(keyspaceName)
        .build();

        RestoreTableResponse response =
keyClient.restoreTable(restoreTableRequest);
        System.out.println("The ARN of the restored table is " +
response.restoredTableARN());

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

- Einzelheiten zur API finden Sie [RestoreTable](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu [GitHub](#). Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

suspend fun restoreTable(
    keyspaceName: String?,
    utc: ZonedDateTime,
) {
    // Create an aws.smithy.kotlin.runtime.time.Instant value.
    val timeStamp =
        aws.smithy.kotlin.runtime.time
            .Instant(utc.toInstant())
    val restoreTableRequest =
        RestoreTableRequest {
            restoreTimestamp = timeStamp
            sourceTableName = "MovieKotlin"
            targetKeyspaceName = keyspaceName
        }
}

```



```

        targetTableName = "MovieRestore"
        sourceKeyspaceName = keyspaceName
    }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        val response = keyClient.restoreTable(restoreTableRequest)
        println("The ARN of the restored table is ${response.restoredTableArn}")
    }
}

```

- Einzelheiten zur API finden Sie [RestoreTable](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```

class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

```

```

def restore_table(self, restore_timestamp):
    """
    Restores the table to a previous point in time. The table is restored
    to a new table in the same keyspace.

    :param restore_timestamp: The point in time to restore the table. This
time
                                must be in UTC format.
    :return: The name of the restored table.
    """
    try:
        restored_table_name = f"{self.table_name}_restored"
        self.keyspaces_client.restore_table(
            sourceKeyspaceName=self.ks_name,
            sourceTableName=self.table_name,
            targetKeyspaceName=self.ks_name,
            targetTableName=restored_table_name,
            restoreTimestamp=restore_timestamp,
        )
    except ClientError as err:
        logger.error(
            "Couldn't restore table %s. Here's why: %s: %s",
            restore_timestamp,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return restored_table_name

```

- Einzelheiten zur API finden Sie [RestoreTable](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **UpdateTable** mit einem AWS SDK oder CLI


Die folgenden Codebeispiele zeigen, wie es verwendet wird `UpdateTable`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Beginnen Sie mit Schlüsselräumen und Tabellen](#)

.NET

AWS SDK for .NET

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/// <summary>
/// Updates the movie table to add a boolean column named watched.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table to change.</param>
/// <returns>The Amazon Resource Name (ARN) of the updated table.</returns>
public async Task<string> UpdateTable(string keyspaceName, string tableName)
{
    var newColumn = new ColumnDefinition { Name = "watched", Type =
"boolean" };
    var request = new UpdateTableRequest
    {
        KeyspaceName = keyspaceName,
        TableName = tableName,
        AddColumns = new List<ColumnDefinition> { newColumn }
    };
    var response = await _amazonKeyspaces.UpdateTableAsync(request);
    return response.ResourceArn;
}
```

- Einzelheiten zur API finden Sie [UpdateTable](#) in der AWS SDK for .NET API-Referenz.

Java

SDK für Java 2.x

 Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
public static void updateTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
    try {
        ColumnDefinition def = ColumnDefinition.builder()
            .name("watched")
            .type("boolean")
            .build();

        UpdateTableRequest tableRequest = UpdateTableRequest.builder()
            .keyspaceName(keySpace)
            .tableName(tableName)
            .addColumnns(def)
            .build();

        keyClient.updateTable(tableRequest);

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

- Einzelheiten zur API finden Sie [UpdateTable](#) in der AWS SDK for Java 2.x API-Referenz.

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
suspend fun updateTable(
    keySpace: String?,
    tableNameVal: String?,
) {
    val def =
        ColumnDefinition {
            name = "watched"
            type = "boolean"
        }


    val tableRequest =
        UpdateTableRequest {
            keyspaceName = keySpace
            tableName = tableNameVal
            addColumns = listOf(def)
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient.updateTable(tableRequest)
    }
}
```

- Einzelheiten zur API finden Sie [UpdateTable](#) in der API-Referenz zum AWS SDK für Kotlin.

Python

SDK für Python (Boto3)

 Note

Es gibt noch mehr dazu. [GitHub](#) Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

    def __init__(self, keyspaces_client):
        """
        :param keyspaces_client: A Boto3 Amazon Keyspaces client.
        """
        self.keyspaces_client = keyspaces_client
        self.ks_name = None
        self.ks_arn = None
        self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

    def update_table(self):
        """
        Updates the schema of the table.

        This example updates a table of movie data by adding a new column
        that tracks whether the movie has been watched.
        """
        try:
            self.keyspaces_client.update_table(
                keyspaceName=self.ks_name,
                tableName=self.table_name,
                addColumns=[{"name": "watched", "type": "boolean"}],
            )
```

```
except ClientError as err:
    logger.error(
        "Couldn't update table %s. Here's why: %s: %s",
        self.table_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise
```

- Einzelheiten zur API finden Sie [UpdateTable](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für Amazon Keyspaces mit AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie gängige Szenarien in Amazon Keyspaces mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben erledigen können, indem Sie mehrere Funktionen in Amazon Keyspaces aufrufen. Jedes Szenario enthält einen Link zu GitHub, über den Sie Anweisungen zur Einrichtung und Ausführung des Codes finden.

Beispiele

- [Erste Schritte mit Amazon Keyspaces-Keyspaces und -Tabellen mithilfe eines SDK AWS](#)

Erste Schritte mit Amazon Keyspaces-Keyspaces und -Tabellen mithilfe eines SDK AWS

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Erstellen Sie einen Schlüsselraum und eine Tabelle. Das Tabellenschema enthält Filmdaten und die point-in-time Wiederherstellung ist aktiviert.
- Connect Sie über eine sichere TLS-Verbindung mit SigV4-Authentifizierung eine Verbindung zum Keyspace her.
- Fragen Sie die Tabelle ab. Filmdaten hinzufügen, abrufen und aktualisieren.

- Aktualisieren Sie die Tabelle. Fügen Sie eine Spalte hinzu, um die angesehenen Filme zu verfolgen.
- Stellen Sie den vorherigen Zustand der Tabelle wieder her und bereinigen Sie die Ressourcen.

.NET

AWS SDK for .NET

Note

Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
global using System.Security.Cryptography.X509Certificates;
global using Amazon.Keyspaces;
global using Amazon.Keyspaces.Model;
global using KeyspacesActions;
global using KeyspacesScenario;
global using Microsoft.Extensions.Configuration;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;
global using Newtonsoft.Json;

namespace KeyspacesBasics;

/// <summary>
/// Amazon Keyspaces (for Apache Cassandra) scenario. Shows some of the basic
/// actions performed with Amazon Keyspaces.
/// </summary>
public class KeyspacesBasics
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
```



```
using var host = Host.CreateDefaultBuilder(args)
    .ConfigureLogging(logging =>
        logging.AddFilter("System", LogLevel.Debug)
            .AddFilter<DebugLoggerProvider>("Microsoft",
                LogLevel.Information)
            .AddFilter<ConsoleLoggerProvider>("Microsoft",
                LogLevel.Trace))
    .ConfigureServices((_, services) =>
        services.AddAWSService<IAmazonKeyspaces>()
        .AddTransient<KeyspacesWrapper>()
        .AddTransient<CassandraWrapper>()
        )
    .Build();

logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
    .CreateLogger<KeyspacesBasics>();

var configuration = new ConfigurationBuilder()
    .SetBasePath(Directory.GetCurrentDirectory())
    .AddJsonFile("settings.json") // Load test settings from .json file.
    .AddJsonFile("settings.local.json",
        true) // Optionally load local settings.
    .Build();

var keyspacesWrapper =
host.Services.GetRequiredService<KeyspacesWrapper>();
var uiMethods = new UiMethods();

var keyspaceName = configuration["KeyspaceName"];
var tableName = configuration["TableName"];

bool success; // Used to track the results of some operations.

uiMethods.DisplayOverview();
uiMethods.PressEnter();

// Create the keyspace.
var keyspaceArn = await keyspacesWrapper.CreateKeyspace(keyspaceName);

// Wait for the keyspace to be available. GetKeyspace results in a
// resource not found error until it is ready for use.
try
{
    var getKeyspaceArn = "";
```

```
        Console.WriteLine($"Created {keyspaceName}. Waiting for it to become
available. ");
        do
        {
            getKeyspaceArn = await
keyspacesWrapper.GetKeyspace(keyspaceName);
            Console.WriteLine(". ");
        } while (getKeyspaceArn != keyspaceArn);
    }
    catch (ResourceNotFoundException)
    {
        Console.WriteLine("Waiting for keyspace to be created.");
    }

    Console.WriteLine($"\\nThe keyspace {keyspaceName} is ready for use.");

    uiMethods.PressEnter();

    // Create the table.
    // First define the schema.
    var allColumns = new List<ColumnDefinition>
    {
        new ColumnDefinition { Name = "title", Type = "text" },
        new ColumnDefinition { Name = "year", Type = "int" },
        new ColumnDefinition { Name = "release_date", Type = "timestamp" },
        new ColumnDefinition { Name = "plot", Type = "text" },
    };

    var partitionKeys = new List<PartitionKey>
    {
        new PartitionKey { Name = "year", },
        new PartitionKey { Name = "title" },
    };

    var tableSchema = new SchemaDefinition
    {
        AllColumns = allColumns,
        PartitionKeys = partitionKeys,
    };

    var tableArn = await keyspacesWrapper.CreateTable(keyspaceName,
tableSchema, tableName);

    // Wait for the table to be active.
```

```
try
{
    var resp = new GetTableResponse();
    Console.WriteLine("Waiting for the new table to be active. ");
    do
    {
        try
        {
            resp = await keyspacesWrapper.GetTable(keyspaceName,
tableName);
            Console.WriteLine(".");
        }
        catch (ResourceNotFoundException)
        {
            Console.WriteLine(".");
        }
    } while (resp.Status != TableStatus.ACTIVE);

    // Display the table's schema.
    Console.WriteLine($"\\nTable {tableName} has been created in
{keyspaceName}");
    Console.WriteLine("Let's take a look at the schema.");
    uiMethods.DisplayTitle("All columns");
    resp.SchemaDefinition.AllColumns.ForEach(column =>
    {
        Console.WriteLine($"{column.Name, -40}\\t{column.Type, -20}");
    });

    uiMethods.DisplayTitle("Cluster keys");
    resp.SchemaDefinition.ClusteringKeys.ForEach(clusterKey =>
    {
        Console.WriteLine($"{clusterKey.Name, -40}\\t{clusterKey.OrderBy, -20}");
    });

    uiMethods.DisplayTitle("Partition keys");
    resp.SchemaDefinition.PartitionKeys.ForEach(partitionKey =>
    {
        Console.WriteLine($"{partitionKey.Name}");
    });

    uiMethods.PressEnter();
}
catch (ResourceNotFoundException ex)
```

```
{
    Console.WriteLine($"Error: {ex.Message}");
}

// Access Apache Cassandra using the Cassandra drive for C#.
var cassandraWrapper =
host.Services.GetRequiredService<CassandraWrapper>();
var movieFilePath = configuration["MovieFile"];

Console.WriteLine("Let's add some movies to the table we created.");
var inserted = await cassandraWrapper.InsertIntoMovieTable(keyspaceName,
tableName, movieFilePath);

uiMethods.PressEnter();

Console.WriteLine("Added the following movies to the table:");
var rows = await cassandraWrapper.GetMovies(keyspaceName, tableName);
uiMethods.DisplayTitle("All Movies");

foreach (var row in rows)
{
    var title = row.GetValue<string>("title");
    var year = row.GetValue<int>("year");
    var plot = row.GetValue<string>("plot");
    var release_date = row.GetValue<DateTime>("release_date");
    Console.WriteLine($"{release_date}\t{title}\t{year}\n{plot}");
    Console.WriteLine(uiMethods.SepBar);
}

// Update the table schema
uiMethods.DisplayTitle("Update table schema");
Console.WriteLine("Now we will update the table to add a boolean field
called watched.");

// First save the current time as a UTC Date so the original
// table can be restored later.
var timeChanged = DateTime.UtcNow;

// Now update the schema.
var resourceArn = await keyspacesWrapper.UpdateTable(keyspaceName,
tableName);
uiMethods.PressEnter();

Console.WriteLine("Now let's mark some of the movies as watched.");
```

```
// Pick some files to mark as watched.
var movieToWatch = rows[2].GetValue<string>("title");
var watchedMovieYear = rows[2].GetValue<int>("year");
var changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[6].GetValue<string>("title");
watchedMovieYear = rows[6].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[9].GetValue<string>("title");
watchedMovieYear = rows[9].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[10].GetValue<string>("title");
watchedMovieYear = rows[10].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[13].GetValue<string>("title");
watchedMovieYear = rows[13].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

uiMethods.DisplayTitle("Watched movies");
Console.WriteLine("These movies have been marked as watched:");
rows = await cassandraWrapper.GetWatchedMovies(keyspaceName, tableName);
foreach (var row in rows)
{
    var title = row.GetValue<string>("title");
    var year = row.GetValue<int>("year");
    Console.WriteLine($"{title, -40}\t{year, 8}");
}
uiMethods.PressEnter();

Console.WriteLine("We can restore the table to its previous state but
that can take up to 20 minutes to complete.");
string answer;
do
{
    Console.WriteLine("Do you want to restore the table? (y/n)");
```

```
        answer = Console.ReadLine();
    } while (answer.ToLower() != "y" && answer.ToLower() != "n");

    if (answer == "y")
    {
        var restoredTableName = $"{tableName}_restored";
        var restoredTableArn = await keyspacesWrapper.RestoreTable(
            keyspaceName,
            tableName,
            restoredTableName,
            timeChanged);
        // Loop and call GetTable until the table is gone. Once it has been
        // deleted completely, GetTable will raise a
ResourceNotFoundException.
        bool wasRestored = false;

        try
        {
            do
            {
                var resp = await keyspacesWrapper.GetTable(keyspaceName,
restoredTableName);
                wasRestored = (resp.Status == TableStatus.ACTIVE);
            } while (!wasRestored);
        }
        catch (ResourceNotFoundException)
        {
            // If the restored table raised an error, it isn't
            // ready yet.
            Console.WriteLine(".");
        }
    }

    uiMethods.DisplayTitle("Clean up resources.");

    // Delete the table.
    success = await keyspacesWrapper.DeleteTable(keyspaceName, tableName);

    Console.WriteLine($"Table {tableName} successfully deleted from
{keyspaceName}.");
    Console.WriteLine("Waiting for the table to be removed completely. ");

    // Loop and call GetTable until the table is gone. Once it has been
    // deleted completely, GetTable will raise a ResourceNotFoundException.
```

```
        bool wasDeleted = false;

        try
        {
            do
            {
                var resp = await keyspacesWrapper.GetTable(keyspaceName,
tableName);
            } while (!wasDeleted);
        }
        catch (ResourceNotFoundException ex)
        {
            wasDeleted = true;
            Console.WriteLine($"{ex.Message} indicates that the table has been
deleted.");
        }

        // Delete the keyspace.
        success = await keyspacesWrapper.DeleteKeyspace(keyspaceName);
        Console.WriteLine("The keyspace has been deleted and the demo is now
complete.");
    }
}
```

```
namespace KeyspacesActions;

/// <summary>
/// Performs Amazon Keyspaces (for Apache Cassandra) actions.
/// </summary>
public class KeyspacesWrapper
{
    private readonly IAmazonKeyspaces _amazonKeyspaces;

    /// <summary>
    /// Constructor for the KeyspaceWrapper.
    /// </summary>
    /// <param name="amazonKeyspaces">An Amazon Keyspaces client object.</param>
    public KeyspacesWrapper(IAmazonKeyspaces amazonKeyspaces)
    {
        _amazonKeyspaces = amazonKeyspaces;
    }
}
```

```
/// <summary>
/// Create a new keyspace.
/// </summary>
/// <param name="keyspaceName">The name for the new keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the new keyspace.</returns>
public async Task<string> CreateKeyspace(string keyspaceName)
{
    var response =
        await _amazonKeyspaces.CreateKeyspaceAsync(
            new CreateKeyspaceRequest { KeyspaceName = keyspaceName });
    return response.ResourceArn;
}

/// <summary>
/// Create a new Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace where the table will be
created.</param>
/// <param name="schema">The schema for the new table.</param>
/// <param name="tableName">The name of the new table.</param>
/// <returns>The Amazon Resource Name (ARN) of the new table.</returns>
public async Task<string> CreateTable(string keyspaceName, SchemaDefinition
schema, string tableName)
{
    var request = new CreateTableRequest
    {
        KeyspaceName = keyspaceName,
        SchemaDefinition = schema,
        TableName = tableName,
        PointInTimeRecovery = new PointInTimeRecovery { Status =
PointInTimeRecoveryStatus.ENABLED }
    };

    var response = await _amazonKeyspaces.CreateTableAsync(request);
    return response.ResourceArn;
}

/// <summary>
/// Delete an existing keyspace.
/// </summary>
/// <param name="keyspaceName"></param>
```



```
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteKeyspace(string keyspaceName)
{
    var response = await _amazonKeyspaces.DeleteKeyspaceAsync(
        new DeleteKeyspaceRequest { KeyspaceName = keyspaceName });
    return response.HttpStatusCode == HttpStatusCode.OK;
}

/// <summary>
/// Delete an Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteTable(string keyspaceName, string tableName)
{
    var response = await _amazonKeyspaces.DeleteTableAsync(
        new DeleteTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
    return response.HttpStatusCode == HttpStatusCode.OK;
}

/// <summary>
/// Get data about a keyspace.
/// </summary>
/// <param name="keyspaceName">The name of the keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the keyspace.</returns>
public async Task<string> GetKeyspace(string keyspaceName)
{
    var response = await _amazonKeyspaces.GetKeyspaceAsync(
        new GetKeyspaceRequest { KeyspaceName = keyspaceName });
    return response.ResourceArn;
}

/// <summary>
/// Get information about an Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the Amazon Keyspaces table.</param>
/// <returns>The response containing data about the table.</returns>
```

```
public async Task<GetTableResponse> GetTable(string keyspaceName, string
tableName)
{
    var response = await _amazonKeyspaces.GetTableAsync(
        new GetTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
    return response;
}

/// <summary>
/// Lists all keyspaces for the account.
/// </summary>
/// <returns>Async task.</returns>
public async Task ListKeyspaces()
{
    var paginator = _amazonKeyspaces.Paginators.ListKeyspaces(new
ListKeyspacesRequest());

    Console.WriteLine("{0, -30}\t{1}", "Keyspace name", "Keyspace ARN");
    Console.WriteLine(new string('-', Console.WindowWidth));
    await foreach (var keyspace in paginator.Keyspaces)
    {
        Console.WriteLine($"{keyspace.KeyspaceName, -30}\t{keyspace.ResourceArn}");
    }
}

/// <summary>
/// Lists the Amazon Keyspaces tables in a keyspace.
/// </summary>
/// <param name="keyspaceName">The name of the keyspace.</param>
/// <returns>A list of TableSummary objects.</returns>
public async Task<List<TableSummary>> ListTables(string keyspaceName)
{
    var response = await _amazonKeyspaces.ListTablesAsync(new
ListTablesRequest { KeyspaceName = keyspaceName });
    response.Tables.ForEach(table =>
    {
        Console.WriteLine($"{table.KeyspaceName}\t{table.TableName}\t{table.ResourceArn}");
    });
}
```

```

        return response.Tables;
    }

    /// <summary>
    /// Restores the specified table to the specified point in time.
    /// </summary>
    /// <param name="keyspaceName">The keyspace containing the table.</param>
    /// <param name="tableName">The name of the table to restore.</param>
    /// <param name="timestamp">The time to which the table will be restored.</
param>
    /// <returns>The Amazon Resource Name (ARN) of the restored table.</returns>
    public async Task<string> RestoreTable(string keyspaceName, string tableName,
string restoredTableName, DateTime timestamp)
    {
        var request = new RestoreTableRequest
        {
            RestoreTimestamp = timestamp,
            SourceKeyspaceName = keyspaceName,
            SourceTableName = tableName,
            TargetKeyspaceName = keyspaceName,
            TargetTableName = restoredTableName
        };

        var response = await _amazonKeyspaces.RestoreTableAsync(request);
        return response.RestoredTableARN;
    }

    /// <summary>
    /// Updates the movie table to add a boolean column named watched.
    /// </summary>
    /// <param name="keyspaceName">The keyspace containing the table.</param>
    /// <param name="tableName">The name of the table to change.</param>
    /// <returns>The Amazon Resource Name (ARN) of the updated table.</returns>
    public async Task<string> UpdateTable(string keyspaceName, string tableName)
    {
        var newColumn = new ColumnDefinition { Name = "watched", Type =
"boolean" };
        var request = new UpdateTableRequest
        {
            KeyspaceName = keyspaceName,
            TableName = tableName,
            AddColumns = new List<ColumnDefinition> { newColumn }
        };
    }

```

```
    };  
    var response = await _amazonKeyspaces.UpdateTableAsync(request);  
    return response.ResourceArn;  
  }  
}
```

```
using System.Net;  
using Cassandra;  
  
namespace KeyspacesScenario;  
  
/// <summary>  
/// Class to perform CRUD methods on an Amazon Keyspaces (for Apache Cassandra)  
/// database.  
///  
/// NOTE: This sample uses a plain text authenticator for example purposes only.  
/// Recommended best practice is to use a SigV4 authentication plugin, if  
/// available.  
/// </summary>  
public class CassandraWrapper  
{  
    private readonly IConfiguration _configuration;  
    private readonly string _localPathToFile;  
    private const string _certLocation = "https://certs.secureserver.net/  
repository/sf-class2-root.crt";  
    private const string _certFileName = "sf-class2-root.crt";  
    private readonly X509Certificate2Collection _certCollection;  
    private X509Certificate2 _amazoncert;  
    private Cluster _cluster;  
  
    // User name and password for the service.  
    private string _userName = null!;  
    private string _pwd = null!;  
  
    public CassandraWrapper()  
    {  
        _configuration = new ConfigurationBuilder()  
            .SetBasePath(Directory.GetCurrentDirectory())  
            .AddJsonFile("settings.json") // Load test settings from .json file.  
            .AddJsonFile("settings.local.json",
```

```
        true) // Optionally load local settings.
        .Build();

    _localPathToFile = Path.GetTempPath();

    // Get the Starfield digital certificate and save it locally.
    var client = new WebClient();
    client.DownloadFile(_certLocation, $"{_localPathToFile}/
{_certFileName}");

    //var httpClient = new HttpClient();
    //var httpResult = httpClient.Get(fileUrl);
    //using var resultStream = await httpResult.Content.ReadAsStreamAsync();
    //using var fileStream = File.Create(pathToSave);
    //resultStream.CopyTo(fileStream);

    _certCollection = new X509Certificate2Collection();
    _amazoncert = new X509Certificate2($"{_localPathToFile}/
{_certFileName}");

    // Get the user name and password stored in the configuration file.
    _userName = _configuration["UserName"]!;
    _pwd = _configuration["Password"]!;

    // For a list of Service Endpoints for Amazon Keyspaces, see:
    // https://docs.aws.amazon.com/keyspaces/latest/devguide/
programmatic.endpoints.html
    var awsEndpoint = _configuration["ServiceEndpoint"];

    _cluster = Cluster.Builder()
        .AddContactPoints(awsEndpoint)
        .WithPort(9142)
        .WithAuthProvider(new PlainTextAuthProvider(_userName, _pwd))
        .WithSSL(new SSLOptions().SetCertificateCollection(_certCollection))
        .WithQueryOptions(
            new QueryOptions()
                .SetConsistencyLevel(ConsistencyLevel.LocalQuorum)
                .SetSerialConsistencyLevel(ConsistencyLevel.LocalSerial))
        .Build();
}

///  

/// Loads the contents of a JSON file into a list of movies to be  

/// added to the Apache Cassandra table.
```

```
    /// </summary>
    /// <param name="movieFileName">The full path to the JSON file.</param>
    /// <returns>A list of movie objects.</returns>
    public List<Movie> ImportMoviesFromJson(string movieFileName, int numToImport
= 0)
    {
        if (!File.Exists(movieFileName))
        {
            return null!;
        }

        using var sr = new StreamReader(movieFileName);
        string json = sr.ReadToEnd();

        var allMovies = JsonConvert.DeserializeObject<List<Movie>>(json);

        // If numToImport = 0, return all movies in the collection.
        if (numToImport == 0)
        {
            // Now return the entire list of movies.
            return allMovies;
        }
        else
        {
            // Now return the first numToImport entries.
            return allMovies.GetRange(0, numToImport);
        }
    }

    /// <summary>
    /// Insert movies into the movie table.
    /// </summary>
    /// <param name="keyspaceName">The keyspace containing the table.</param>
    /// <param name="movieTableName">The Amazon Keyspaces table.</param>
    /// <param name="movieFilePath">The path to the resource file containing
    /// movie data to insert into the table.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> InsertIntoMovieTable(string keyspaceName, string
movieTableName, string movieFilePath, int numToImport = 20)
    {
        // Get some movie data from the movies.json file
        var movies = ImportMoviesFromJson(movieFilePath, numToImport);

        var session = _cluster.Connect(keyspaceName);
```

```
    string insertCql;

    RowSet rs;

    // Now we insert the numToImport movies into the table.
    foreach (var movie in movies)
    {
        // Escape single quote characters in the plot.
        insertCql = $"INSERT INTO {keyspaceName}.{movieTableName}
(title, year, release_date, plot) values({${movie.Title}}$, {movie.Year},
'{movie.Info.Release_Date.ToString("yyyy-MM-dd")}', ${movie.Info.Plot}$)";
        rs = await session.ExecuteAsync(new SimpleStatement(insertCql));
    }

    return true;
}

/// <summary>
/// Gets all of the movies in the movies table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table.</param>
/// <returns>A list of row objects containing movie data.</returns>
public async Task<List<Row>> GetMovies(string keyspaceName, string tableName)
{
    var session = _cluster.Connect();
    RowSet rs;
    try
    {
        rs = await session.ExecuteAsync(new SimpleStatement($"SELECT * FROM
{keyspaceName}.{tableName}"));

        // Extract the row data from the returned RowSet.
        var rows = rs.GetRows().ToList();
        return rows;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null!;
    }
}
```

```

    /// <summary>
    /// Mark a movie in the movie table as watched.
    /// </summary>
    /// <param name="keyspaceName">The keyspace containing the table.</param>
    /// <param name="tableName">The name of the table.</param>
    /// <param name="title">The title of the movie to mark as watched.</param>
    /// <param name="year">The year the movie was released.</param>
    /// <returns>A set of rows containing the changed data.</returns>
    public async Task<List<Row>> MarkMovieAsWatched(string keyspaceName, string
tableName, string title, int year)
    {
        var session = _cluster.Connect();
        string updateCql = $"UPDATE {keyspaceName}.{tableName} SET watched=true
WHERE title = ${title} AND year = {year}";
        var rs = await session.ExecuteAsync(new SimpleStatement(updateCql));
        var rows = rs.GetRows().ToList();
        return rows;
    }

    /// <summary>
    /// Retrieve the movies in the movies table where watched is true.
    /// </summary>
    /// <param name="keyspaceName">The keyspace containing the table.</param>
    /// <param name="tableName">The name of the table.</param>
    /// <returns>A list of row objects containing information about movies
    /// where watched is true.</returns>
    public async Task<List<Row>> GetWatchedMovies(string keyspaceName, string
tableName)
    {
        var session = _cluster.Connect();
        RowSet rs;
        try
        {
            rs = await session.ExecuteAsync(new SimpleStatement($"SELECT
title, year, plot FROM {keyspaceName}.{tableName} WHERE watched = true ALLOW
FILTERING"));

            // Extract the row data from the returned RowSet.
            var rows = rs.GetRows().ToList();
            return rows;
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }

```



```
        return null!;  
    }  
}  
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK for .NET -API-Referenz.
 - [CreateKeyspace](#)
 - [CreateTable](#)
 - [DeleteKeyspace](#)
 - [DeleteTable](#)
 - [GetKeyspace](#)
 - [GetTable](#)
 - [ListKeyspaces](#)
 - [ListTables](#)
 - [RestoreTable](#)
 - [UpdateTable](#)

Java

SDK für Java 2.x

Note

Es gibt noch mehr GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/**  
 * Before running this Java (v2) code example, set up your development  
 * environment, including your credentials.  
 *  
 * For more information, see the following documentation topic:  
 *  
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html  
 */
```

```
*
* Before running this Java code example, you must create a
* Java keystore (JKS) file and place it in your project's resources folder.
*
* This file is a secure file format used to hold certificate information for
* Java applications. This is required to make a connection to Amazon Keyspaces.
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/keyspaces/latest/devguide/using\_java\_driver.html
*
* This Java example performs the following tasks:
*
* 1. Create a keyspace.
* 2. Check for keyspace existence.
* 3. List keyspaces using a paginator.
* 4. Create a table with a simple movie data schema and enable point-in-time
* recovery.
* 5. Check for the table to be in an Active state.
* 6. List all tables in the keyspace.
* 7. Use a Cassandra driver to insert some records into the Movie table.
* 8. Get all records from the Movie table.
* 9. Get a specific Movie.
* 10. Get a UTC timestamp for the current time.
* 11. Update the table schema to add a 'watched' Boolean column.
* 12. Update an item as watched.
* 13. Query for items with watched = True.
* 14. Restore the table back to the previous state using the timestamp.
* 15. Check for completion of the restore action.
* 16. Delete the table.
* 17. Confirm that both tables are deleted.
* 18. Delete the keyspace.
*/

public class ScenarioKeyspaces {
    public static final String DASHES = new String(new char[80]).replace("\0",
"-");

    /*
    * Usage:
    * fileName - The name of the JSON file that contains movie data. (Get this
file
    * from the GitHub repo at resources/sample_file.)
    * keyspaceName - The name of the keyspace to create.
    */
}
```

```
public static void main(String[] args) throws InterruptedException,
IOException {
    String fileName = "<Replace with the JSON file that contains movie
data>";
    String keyspaceName = "<Replace with the name of the keyspace to
create>";
    String titleUpdate = "The Family";
    int yearUpdate = 2013;
    String tableName = "Movie";
    String tableNameRestore = "MovieRestore";
    Region region = Region.US_EAST_1;
    KeyspacesClient keyClient = KeyspacesClient.builder()
        .region(region)
        .build();

    DriverConfigLoader loader =
DriverConfigLoader.fromClasspath("application.conf");
    CqlSession session = CqlSession.builder()
        .withConfigLoader(loader)
        .build();

    System.out.println(DASHES);
    System.out.println("Welcome to the Amazon Keyspaces example scenario.");
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("1. Create a keyspace.");
    createKeySpace(keyClient, keyspaceName);
    System.out.println(DASHES);

    System.out.println(DASHES);
    Thread.sleep(5000);
    System.out.println("2. Check for keyspace existence.");
    checkKeyspaceExistence(keyClient, keyspaceName);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("3. List keyspaces using a paginator.");
    listKeyspacesPaginator(keyClient);
    System.out.println(DASHES);

    System.out.println(DASHES);
    System.out.println("4. Create a table with a simple movie data schema and
enable point-in-time recovery.");
```

```
createTable(keyClient, keyspaceName, tableName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("5. Check for the table to be in an Active state.");
Thread.sleep(6000);
checkTable(keyClient, keyspaceName, tableName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("6. List all tables in the keyspace.");
listTables(keyClient, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("7. Use a Cassandra driver to insert some records into
the Movie table.");
Thread.sleep(6000);
loadData(session, fileName, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("8. Get all records from the Movie table.");
getMovieData(session, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("9. Get a specific Movie.");
getSpecificMovie(session, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("10. Get a UTC timestamp for the current time.");
ZonedDateTime utc = ZonedDateTime.now(ZoneOffset.UTC);
System.out.println("DATETIME = " + Date.from(utc.toInstant()));
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("11. Update the table schema to add a watched Boolean
column.");
updateTable(keyClient, keyspaceName, tableName);
System.out.println(DASHES);

System.out.println(DASHES);
```

```
System.out.println("12. Update an item as watched.");
Thread.sleep(10000); // Wait 10 secs for the update.
updateRecord(session, keyspaceName, titleUpdate, yearUpdate);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("13. Query for items with watched = True.");
getWatchedData(session, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("14. Restore the table back to the previous state
using the timestamp.");
System.out.println("Note that the restore operation can take up to 20
minutes.");
restoreTable(keyClient, keyspaceName, utc);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("15. Check for completion of the restore action.");
Thread.sleep(5000);
checkRestoredTable(keyClient, keyspaceName, "MovieRestore");
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("16. Delete both tables.");
deleteTable(keyClient, keyspaceName, tableName);
deleteTable(keyClient, keyspaceName, tableNameRestore);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("17. Confirm that both tables are deleted.");
checkTableDelete(keyClient, keyspaceName, tableName);
checkTableDelete(keyClient, keyspaceName, tableNameRestore);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("18. Delete the keyspace.");
deleteKeyspace(keyClient, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("The scenario has completed successfully.");
System.out.println(DASHES);
```

```
    }

    public static void deleteKeyspace(KeyspacesClient keyClient, String
keyspaceName) {
        try {
            DeleteKeyspaceRequest deleteKeyspaceRequest =
DeleteKeyspaceRequest.builder()
                .keyspaceName(keyspaceName)
                .build();

            keyClient.deleteKeyspace(deleteKeyspaceRequest);

        } catch (KeyspacesException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void checkTableDelete(KeyspacesClient keyClient, String
keyspaceName, String tableName)
        throws InterruptedException {
        try {
            String status;
            GetTableResponse response;
            GetTableRequest tableRequest = GetTableRequest.builder()
                .keyspaceName(keyspaceName)
                .tableName(tableName)
                .build();

            // Keep looping until table cannot be found and a
ResourceNotFoundException is
            // thrown.
            while (true) {
                response = keyClient.getTable(tableRequest);
                status = response.statusAsString();
                System.out.println(". The table status is " + status);
                Thread.sleep(500);
            }

        } catch (ResourceNotFoundException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        System.out.println("The table is deleted");
    }
}
```

```
public static void deleteTable(KeyspacesClient keyClient, String
keyspaceName, String tableName) {
    try {
        DeleteTableRequest tableRequest = DeleteTableRequest.builder()
            .keyspaceName(keyspaceName)
            .tableName(tableName)
            .build();

        keyClient.deleteTable(tableRequest);

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void checkRestoredTable(KeyspacesClient keyClient, String
keyspaceName, String tableName)
    throws InterruptedException {
    try {
        boolean tableStatus = false;
        String status;
        GetTableResponse response = null;
        GetTableRequest tableRequest = GetTableRequest.builder()
            .keyspaceName(keyspaceName)
            .tableName(tableName)
            .build();

        while (!tableStatus) {
            response = keyClient.getTable(tableRequest);
            status = response.statusAsString();
            System.out.println("The table status is " + status);

            if (status.compareTo("ACTIVE") == 0) {
                tableStatus = true;
            }
            Thread.sleep(500);
        }

        List<ColumnDefinition> cols =
response.schemaDefinition().allColumns();
        for (ColumnDefinition def : cols) {
            System.out.println("The column name is " + def.name());
        }
    }
}
```

```
        System.out.println("The column type is " + def.type());
    }

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void restoreTable(KeyspacesClient keyClient, String
keyspaceName, ZonedDateTime utc) {
    try {
        Instant myTime = utc.toInstant();
        RestoreTableRequest restoreTableRequest =
RestoreTableRequest.builder()
            .restoreTimestamp(myTime)
            .sourceTableName("Movie")
            .targetKeyspaceName(keyspaceName)
            .targetTableName("MovieRestore")
            .sourceKeyspaceName(keyspaceName)
            .build();

        RestoreTableResponse response =
keyClient.restoreTable(restoreTableRequest);
        System.out.println("The ARN of the restored table is " +
response.restoredTableARN());

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void getWatchedData(CqlSession session, String keyspaceName) {
    ResultSet resultSet = session
        .execute("SELECT * FROM \"" + keyspaceName + "\".\"Movie\" WHERE
watched = true ALLOW FILTERING;");
    resultSet.forEach(item -> {
        System.out.println("The Movie title is " + item.getString("title"));
        System.out.println("The Movie year is " + item.getInt("year"));
        System.out.println("The plot is " + item.getString("plot"));
    });
}
```



```
public static void updateRecord(CqlSession session, String keySpace, String
titleUpdate, int yearUpdate) {
    String sqlStatement = "UPDATE \"" + keySpace
        + "\".\"Movie\" SET watched=true WHERE title = :k0 AND year
= :k1;";
    BatchStatementBuilder builder =
BatchStatement.builder(DefaultBatchType.UNLOGGED);
    builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM);
    PreparedStatement preparedStatement = session.prepare(sqlStatement);
    builder.addStatement(preparedStatement.boundStatementBuilder()
        .setString("k0", titleUpdate)
        .setInt("k1", yearUpdate)
        .build());

    BatchStatement batchStatement = builder.build();
    session.execute(batchStatement);
}

public static void updateTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
    try {
        ColumnDefinition def = ColumnDefinition.builder()
            .name("watched")
            .type("boolean")
            .build();

        UpdateTableRequest tableRequest = UpdateTableRequest.builder()
            .keyspaceName(keySpace)
            .tableName(tableName)
            .addColumnns(def)
            .build();

        keyClient.updateTable(tableRequest);

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void getSpecificMovie(CqlSession session, String keyspaceName)
{
    ResultSet resultSet = session.execute(
```

```

        "SELECT * FROM \"" + keyspaceName + "\".\"Movie\" WHERE title =
'The Family' ALLOW FILTERING ;");
    resultSet.forEach(item -> {
        System.out.println("The Movie title is " + item.getString("title"));
        System.out.println("The Movie year is " + item.getInt("year"));
        System.out.println("The plot is " + item.getString("plot"));
    });
}

// Get records from the Movie table.
public static void getMovieData(CqlSession session, String keyspaceName) {
    ResultSet resultSet = session.execute("SELECT * FROM \"" + keyspaceName +
"\".\"Movie\"");
    resultSet.forEach(item -> {
        System.out.println("The Movie title is " + item.getString("title"));
        System.out.println("The Movie year is " + item.getInt("year"));
        System.out.println("The plot is " + item.getString("plot"));
    });
}

// Load data into the table.
public static void loadData(CqlSession session, String fileName, String
keySpace) throws IOException {
    String sqlStatement = "INSERT INTO \"" + keySpace + "\".\"Movie\" (title,
year, plot) values (:k0, :k1, :k2)";
    JsonParser parser = new JsonFactory().createParser(new File(fileName));
    com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
    Iterator<JsonNode> iter = rootNode.iterator();
    ObjectNode currentNode;
    int t = 0;
    while (iter.hasNext()) {

        // Add 20 movies to the table.
        if (t == 20)
            break;
        currentNode = (ObjectNode) iter.next();

        int year = currentNode.path("year").asInt();
        String title = currentNode.path("title").asText();
        String plot = currentNode.path("info").path("plot").toString();

        // Insert the data into the Amazon Keyspaces table.

```

```
        BatchStatementBuilder builder =
BatchStatement.builder(DefaultBatchType.UNLOGGED);
        builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM);
        PreparedStatement preparedStatement = session.prepare(sqlStatement);
        builder.addStatement(preparedStatement.boundStatementBuilder()
                .setString("k0", title)
                .setInt("k1", year)
                .setString("k2", plot)
                .build());

        BatchStatement batchStatement = builder.build();
        session.execute(batchStatement);
        t++;
    }

    System.out.println("You have added " + t + " records successfully!");
}

public static void listTables(KeyspacesClient keyClient, String keyspaceName)
{
    try {
        ListTablesRequest tablesRequest = ListTablesRequest.builder()
                .keyspaceName(keyspaceName)
                .build();

        ListTablesIterable listRes =
keyClient.listTablesPaginator(tablesRequest);
        listRes.stream()
                .flatMap(r -> r.tables().stream())
                .forEach(content -> System.out.println(" ARN: " +
content.resourceArn() +
                " Table name: " + content.tableName()));

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void checkTable(KeyspacesClient keyClient, String keyspaceName,
String tableName)
    throws InterruptedException {
    try {
        boolean tableStatus = false;
```

```
String status;
GetTableResponse response = null;
GetTableRequest tableRequest = GetTableRequest.builder()
    .keyspaceName(keyspaceName)
    .tableName(tableName)
    .build();

while (!tableStatus) {
    response = keyClient.getTable(tableRequest);
    status = response.statusAsString();
    System.out.println(". The table status is " + status);

    if (status.compareTo("ACTIVE") == 0) {
        tableStatus = true;
    }
    Thread.sleep(500);
}

List<ColumnDefinition> cols =
response.schemaDefinition().allColumns();
for (ColumnDefinition def : cols) {
    System.out.println("The column name is " + def.name());
    System.out.println("The column type is " + def.type());
}

} catch (KeyspacesException e) {
    System.err.println(e.awsErrorDetails().errorMessage());
    System.exit(1);
}
}

public static void createTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
    try {
        // Set the columns.
        ColumnDefinition defTitle = ColumnDefinition.builder()
            .name("title")
            .type("text")
            .build();

        ColumnDefinition defYear = ColumnDefinition.builder()
            .name("year")
            .type("int")
            .build();
```

```
ColumnDefinition defReleaseDate = ColumnDefinition.builder()
    .name("release_date")
    .type("timestamp")
    .build();

ColumnDefinition defPlot = ColumnDefinition.builder()
    .name("plot")
    .type("text")
    .build();

List<ColumnDefinition> collist = new ArrayList<>();
collist.add(defTitle);
collist.add(defYear);
collist.add(defReleaseDate);
collist.add(defPlot);

// Set the keys.
PartitionKey yearKey = PartitionKey.builder()
    .name("year")
    .build();

PartitionKey titleKey = PartitionKey.builder()
    .name("title")
    .build();

List<PartitionKey> keyList = new ArrayList<>();
keyList.add(yearKey);
keyList.add(titleKey);

SchemaDefinition schemaDefinition = SchemaDefinition.builder()
    .partitionKeys(keyList)
    .allColumns(collist)
    .build();

PointInTimeRecovery timeRecovery = PointInTimeRecovery.builder()
    .status(PointInTimeRecoveryStatus.ENABLED)
    .build();

CreateTableRequest tableRequest = CreateTableRequest.builder()
    .keyspaceName(keySpace)
    .tableName(tableName)
    .schemaDefinition(schemaDefinition)
    .pointInTimeRecovery(timeRecovery)
```

```
        .build();

        CreateTableResponse response = keyClient.createTable(tableRequest);
        System.out.println("The table ARN is " + response.resourceArn());

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void listKeyspacesPaginator(KeyspacesClient keyClient) {
    try {
        ListKeyspacesRequest keyspacesRequest =
ListKeyspacesRequest.builder()
            .maxResults(10)
            .build();

        ListKeyspacesIterable listRes =
keyClient.listKeyspacesPaginator(keyspacesRequest);
        listRes.stream()
            .flatMap(r -> r.keyspaces().stream())
            .forEach(content -> System.out.println(" Name: " +
content.keyspaceName()));

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void checkKeyspaceExistence(KeyspacesClient keyClient, String
keyspaceName) {
    try {
        GetKeyspaceRequest keyspaceRequest = GetKeyspaceRequest.builder()
            .keyspaceName(keyspaceName)
            .build();

        GetKeyspaceResponse response =
keyClient.getKeyspace(keyspaceRequest);
        String name = response.keyspaceName();
        System.out.println("The " + name + " KeySpace is ready");

    } catch (KeyspacesException e) {
```

```
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

public static void createKeySpace(KeyspacesClient keyClient, String
keyspaceName) {
    try {
        CreateKeyspaceRequest keyspaceRequest =
CreateKeyspaceRequest.builder()
            .keyspaceName(keyspaceName)
            .build();

        CreateKeyspaceResponse response =
keyClient.createKeyspace(keyspaceRequest);
        System.out.println("The ARN of the KeySpace is " +
response.resourceArn());

    } catch (KeyspacesException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
```

- API-Details finden Sie in den folgenden Themen der AWS SDK for Java 2.x -API-Referenz.
 - [CreateKeyspace](#)
 - [CreateTable](#)
 - [DeleteKeyspace](#)
 - [DeleteTable](#)
 - [GetKeyspace](#)
 - [GetTable](#)
 - [ListKeyspaces](#)
 - [ListTables](#)
 - [RestoreTable](#)
 - [UpdateTable](#)

Kotlin

SDK für Kotlin

Note

Es gibt noch mehr GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

```
/**
```

```
Before running this Kotlin code example, set up your development environment, including your credentials.
```

```
For more information, see the following documentation topic:
```

```
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
```

```
This example uses a secure file format to hold certificate information for Kotlin applications. This is required to make a connection to Amazon Keyspaces. For more information, see the following documentation topic:
```

```
https://docs.aws.amazon.com/keyspaces/latest/devguide/using\_java\_driver.html
```

```
This Kotlin example performs the following tasks:
```

1. Create a keyspace.
2. Check for keyspace existence.
3. List keyspaces using a paginator.
4. Create a table with a simple movie data schema and enable point-in-time recovery.
5. Check for the table to be in an Active state.
6. List all tables in the keyspace.
7. Use a Cassandra driver to insert some records into the Movie table.
8. Get all records from the Movie table.
9. Get a specific Movie.
10. Get a UTC timestamp for the current time.
11. Update the table schema to add a 'watched' Boolean column.
12. Update an item as watched.
13. Query for items with watched = True.
14. Restore the table back to the previous state using the timestamp.


```
15. Check for completion of the restore action.
16. Delete the table.
17. Confirm that both tables are deleted.
18. Delete the keyspace.
*/

/*
Usage:
  fileName - The name of the JSON file that contains movie data. (Get this
file from the GitHub repo at resources/sample_file.)
  keyspaceName - The name of the keyspace to create.
*/
val DASHES: String = String(CharArray(80)).replace("\u0000", "-")

suspend fun main() {
    val fileName = "<Replace with the JSON file that contains movie data>"
    val keyspaceName = "<Replace with the name of the keyspace to create>"
    val titleUpdate = "The Family"
    val yearUpdate = 2013
    val tableName = "MovieKotlin"
    val tableNameRestore = "MovieRestore"

    val loader = DriverConfigLoader.fromClasspath("application.conf")
    val session =
        CqlSession
            .builder()
            .withConfigLoader(loader)
            .build()

    println(DASHES)
    println("Welcome to the Amazon Keyspaces example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. Create a keyspace.")
    createKeySpace(keyspaceName)
    println(DASHES)

    println(DASHES)
    delay(5000)
    println("2. Check for keyspace existence.")
    checkKeyspaceExistence(keyspaceName)
    println(DASHES)
```

```
println(DASHES)
println("3. List keyspaces using a paginator.")
listKeyspacesPaginator()
println(DASHES)

println(DASHES)
println("4. Create a table with a simple movie data schema and enable point-
in-time recovery.")
createTable(keyspaceName, tableName)
println(DASHES)

println(DASHES)
println("5. Check for the table to be in an Active state.")
delay(6000)
checkTable(keyspaceName, tableName)
println(DASHES)

println(DASHES)
println("6. List all tables in the keyspace.")
listTables(keyspaceName)
println(DASHES)

println(DASHES)
println("7. Use a Cassandra driver to insert some records into the Movie
table.")
delay(6000)
loadData(session, fileName, keyspaceName)
println(DASHES)

println(DASHES)
println("8. Get all records from the Movie table.")
getMovieData(session, keyspaceName)
println(DASHES)

println(DASHES)
println("9. Get a specific Movie.")
getSpecificMovie(session, keyspaceName)
println(DASHES)

println(DASHES)
println("10. Get a UTC timestamp for the current time.")
val utc = ZonedDateTime.now(ZoneOffset.UTC)
println("DATETIME = ${Date.from(utc.toInstant())}")
println(DASHES)
```

```
println(DASHES)
println("11. Update the table schema to add a watched Boolean column.")
updateTable(keyspaceName, tableName)
println(DASHES)

println(DASHES)
println("12. Update an item as watched.")
delay(10000) // Wait 10 seconds for the update.
updateRecord(session, keyspaceName, titleUpdate, yearUpdate)
println(DASHES)

println(DASHES)
println("13. Query for items with watched = True.")
getWatchedData(session, keyspaceName)
println(DASHES)

println(DASHES)
println("14. Restore the table back to the previous state using the
timestamp.")
println("Note that the restore operation can take up to 20 minutes.")
restoreTable(keyspaceName, utc)
println(DASHES)

println(DASHES)
println("15. Check for completion of the restore action.")
delay(5000)
checkRestoredTable(keyspaceName, "MovieRestore")
println(DASHES)

println(DASHES)
println("16. Delete both tables.")
deleteTable(keyspaceName, tableName)
deleteTable(keyspaceName, tableNameRestore)
println(DASHES)

println(DASHES)
println("17. Confirm that both tables are deleted.")
checkTableDelete(keyspaceName, tableName)
checkTableDelete(keyspaceName, tableNameRestore)
println(DASHES)

println(DASHES)
println("18. Delete the keyspace.")
```

```
deleteKeyspace(keyspaceName)
println(DASHES)

println(DASHES)
println("The scenario has completed successfully.")
println(DASHES)
}

suspend fun deleteKeyspace(keyspaceNameVal: String?) {
    val deleteKeyspaceRequest =
        DeleteKeyspaceRequest {
            keyspaceName = keyspaceNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient.deleteKeyspace(deleteKeyspaceRequest)
    }
}

suspend fun checkTableDelete(
    keyspaceNameVal: String?,
    tableNameVal: String?,
) {
    var status: String
    var response: GetTableResponse
    val tableRequest =
        GetTableRequest {
            keyspaceName = keyspaceNameVal
            tableName = tableNameVal
        }

    try {
        KeyspacesClient { region = "us-east-1" }.use { keyClient ->
            // Keep looping until the table cannot be found and a
            ResourceNotFoundException is thrown.
            while (true) {
                response = keyClient.getTable(tableRequest)
                status = response.status.toString()
                println(". The table status is $status")
                delay(500)
            }
        }
    } catch (e: ResourceNotFoundException) {
        println(e.message)
    }
}
```

```
    }
    println("The table is deleted")
}

suspend fun deleteTable(
    keyspaceNameVal: String?,
    tableNameVal: String?,
) {
    val tableRequest =
        DeleteTableRequest {
            keyspaceName = keyspaceNameVal
            tableName = tableNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient.deleteTable(tableRequest)
    }
}

suspend fun checkRestoredTable(
    keyspaceNameVal: String?,
    tableNameVal: String?,
) {
    var tableStatus = false
    var status: String
    var response: GetTableResponse? = null

    val tableRequest =
        GetTableRequest {
            keyspaceName = keyspaceNameVal
            tableName = tableNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        while (!tableStatus) {
            response = keyClient.getTable(tableRequest)
            status = response!!.status.toString()
            println("The table status is $status")

            if (status.compareTo("ACTIVE") == 0) {
                tableStatus = true
            }
            delay(500)
        }
    }
}
```

```

        val cols = response!!.schemaDefinition?.allColumns
        if (cols != null) {
            for (def in cols) {
                println("The column name is ${def.name}")
                println("The column type is ${def.type}")
            }
        }
    }
}

suspend fun restoreTable(
    keyspaceName: String?,
    utc: ZonedDateTime,
) {
    // Create an aws.smithy.kotlin.runtime.time.Instant value.
    val timeStamp =
        aws.smithy.kotlin.runtime.time
            .Instant(utc.toInstant())
    val restoreTableRequest =
        RestoreTableRequest {
            restoreTimestamp = timeStamp
            sourceTableName = "MovieKotlin"
            targetKeyspaceName = keyspaceName
            targetTableName = "MovieRestore"
            sourceKeyspaceName = keyspaceName
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        val response = keyClient.restoreTable(restoreTableRequest)
        println("The ARN of the restored table is ${response.restoredTableArn}")
    }
}

fun getWatchedData(
    session: CqlSession,
    keyspaceName: String,
) {
    val resultSet = session.execute("SELECT * FROM \"${keyspaceName}\".
    \"MovieKotlin\" WHERE watched = true ALLOW FILTERING;")
    resultSet.forEach { item: Row ->
        println("The Movie title is ${item.getString("title")}")
        println("The Movie year is ${item.getInt("year")}")
        println("The plot is ${item.getString("plot")}")
    }
}

```

```
    }  
  }  
  
fun updateRecord(  
    session: CqlSession,  
    keySpace: String,  
    titleUpdate: String?,  
    yearUpdate: Int,  
) {  
    val sqlStatement =  
        "UPDATE \"\$keySpace\".\"MovieKotlin\" SET watched=true WHERE title = :k0  
AND year = :k1;"  
    val builder = BatchStatement.builder(DefaultBatchType.UNLOGGED)  
    builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM)  
    val preparedStatement = session.prepare(sqlStatement)  
    builder.addStatement(  
        preparedStatement  
            .boundStatementBuilder()  
            .setString("k0", titleUpdate)  
            .setInt("k1", yearUpdate)  
            .build(),  
    )  
    val batchStatement = builder.build()  
    session.execute(batchStatement)  
}  
  
suspend fun updateTable(  
    keySpace: String?,  
    tableNameVal: String?,  
) {  
    val def =  
        ColumnDefinition {  
            name = "watched"  
            type = "boolean"  
        }  
  
    val tableRequest =  
        UpdateTableRequest {  
            keyspaceName = keySpace  
            tableName = tableNameVal  
            addColumns = listOf(def)  
        }  
  
    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
```

```
        keyClient.updateTable(tableRequest)
    }
}

fun getSpecificMovie(
    session: CqlSession,
    keyspaceName: String,
) {
    val resultSet =
        session.execute("SELECT * FROM \"\$keyspaceName\".\"MovieKotlin\" WHERE
title = 'The Family' ALLOW FILTERING ;")

    resultSet.forEach { item: Row ->
        println("The Movie title is \"${item.getString("title")}\"")
        println("The Movie year is \"${item.getInt("year")}\"")
        println("The plot is \"${item.getString("plot")}\"")
    }
}

// Get records from the Movie table.
fun getMovieData(
    session: CqlSession,
    keyspaceName: String,
) {
    val resultSet = session.execute("SELECT * FROM \"\$keyspaceName\".
\"MovieKotlin\";")
    resultSet.forEach { item: Row ->
        println("The Movie title is \"${item.getString("title")}\"")
        println("The Movie year is \"${item.getInt("year")}\"")
        println("The plot is \"${item.getString("plot")}\"")
    }
}

// Load data into the table.
fun loadData(
    session: CqlSession,
    fileName: String,
    keySpace: String,
) {
    val sqlStatement =
        "INSERT INTO \"\$keySpace\".\"MovieKotlin\" (title, year, plot) values
(:k0, :k1, :k2)"
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
```



```
val iter: Iterator<JsonNode> = rootNode.iterator()
var currentNode: ObjectNode

var t = 0
while (iter.hasNext()) {
    if (t == 50) {
        break
    }

    currentNode = iter.next() as ObjectNode
    val year = currentNode.path("year").asInt()
    val title = currentNode.path("title").asText()
    val info = currentNode.path("info").toString()

    // Insert the data into the Amazon Keyspaces table.
    val builder = BatchStatement.builder(DefaultBatchType.UNLOGGED)
    builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM)
    val preparedStatement: PreparedStatement = session.prepare(sqlStatement)
    builder.addStatement(
        preparedStatement
            .boundStatementBuilder()
            .setString("k0", title)
            .setInt("k1", year)
            .setString("k2", info)
            .build(),
    )

    val batchStatement = builder.build()
    session.execute(batchStatement)
    t++
}

suspend fun listTables(keyspaceNameVal: String?) {
    val tablesRequest =
        ListTablesRequest {
            keyspaceName = keyspaceNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient
            .listTablesPaginated(tablesRequest)
            .transform { it.tables?.forEach { obj -> emit(obj) } }
            .collect { obj ->
```

```

        println(" ARN: ${obj.resourceArn} Table name: ${obj.tableName}")
    }
}

suspend fun checkTable(
    keyspaceNameVal: String?,
    tableNameVal: String?,
) {
    var tableStatus = false
    var status: String
    var response: GetTableResponse? = null

    val tableRequest =
        GetTableRequest {
            keyspaceName = keyspaceNameVal
            tableName = tableNameVal
        }
    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        while (!tableStatus) {
            response = keyClient.getTable(tableRequest)
            status = response!!.status.toString()
            println(". The table status is $status")
            if (status.compareTo("ACTIVE") == 0) {
                tableStatus = true
            }
            delay(500)
        }
        val cols: List<ColumnDefinition>? =
response!!.schemaDefinition?.allColumns
        if (cols != null) {
            for (def in cols) {
                println("The column name is ${def.name}")
                println("The column type is ${def.type}")
            }
        }
    }
}

suspend fun createTable(
    keySpaceVal: String?,
    tableNameVal: String?,
) {
    // Set the columns.

```

```
val defTitle =
    ColumnDefinition {
        name = "title"
        type = "text"
    }

val defYear =
    ColumnDefinition {
        name = "year"
        type = "int"
    }

val defReleaseDate =
    ColumnDefinition {
        name = "release_date"
        type = "timestamp"
    }

val defPlot =
    ColumnDefinition {
        name = "plot"
        type = "text"
    }

val collList = ArrayList<ColumnDefinition>()
collList.add(defTitle)
collList.add(defYear)
collList.add(defReleaseDate)
collList.add(defPlot)

// Set the keys.
val yearKey =
    PartitionKey {
        name = "year"
    }

val titleKey =
    PartitionKey {
        name = "title"
    }

val keyList = ArrayList<PartitionKey>()
keyList.add(yearKey)
keyList.add(titleKey)
```

```
val schemaDefinitionObj =
    SchemaDefinition {
        partitionKeys = keyList
        allColumns = colList
    }

val timeRecovery =
    PointInTimeRecovery {
        status = PointInTimeRecoveryStatus.Enabled
    }

val tableRequest =
    CreateTableRequest {
        keyspaceName = keySpaceVal
        tableName = tableNameVal
        schemaDefinition = schemaDefinitionObj
        pointInTimeRecovery = timeRecovery
    }

KeyspacesClient { region = "us-east-1" }.use { keyClient ->
    val response = keyClient.createTable(tableRequest)
    println("The table ARN is ${response.resourceArn}")
}

suspend fun listKeyspacesPaginator() {
    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        keyClient
            .listKeyspacesPaginated(ListKeyspacesRequest {})
            .transform { it.keyspaces?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name: ${obj.keyspaceName}")
            }
    }
}

suspend fun checkKeyspaceExistence(keyspaceNameVal: String?) {
    val keyspaceRequest =
        GetKeyspaceRequest {
            keyspaceName = keyspaceNameVal
        }
    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
```

```
        val response: GetKeyspaceResponse =
        keyClient.getKeyspace(keyspaceRequest)
        val name = response.keyspaceName
        println("The $name KeySpace is ready")
    }
}

suspend fun createKeySpace(keyspaceNameVal: String) {
    val keyspaceRequest =
        CreateKeyspaceRequest {
            keyspaceName = keyspaceNameVal
        }

    KeyspacesClient { region = "us-east-1" }.use { keyClient ->
        val response = keyClient.createKeyspace(keyspaceRequest)
        println("The ARN of the KeySpace is ${response.resourceArn}")
    }
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Kotlin.
 - [CreateKeyspace](#)
 - [CreateTable](#)
 - [DeleteKeyspace](#)
 - [DeleteTable](#)
 - [GetKeyspace](#)
 - [GetTable](#)
 - [ListKeyspaces](#)
 - [ListTables](#)
 - [RestoreTable](#)
 - [UpdateTable](#)

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-Repository](#) einrichten und ausführen.

Führen Sie ein interaktives Szenario an einer Eingabeaufforderung aus.

```
class KeyspaceScenario:
    """Runs an interactive scenario that shows how to get started using Amazon
    Keyspaces."""

    def __init__(self, ks_wrapper):
        """
        :param ks_wrapper: An object that wraps Amazon Keyspace actions.
        """
        self.ks_wrapper = ks_wrapper

    @demo_func
    def create_keyspace(self):
        """
        1. Creates a keyspace.
        2. Lists up to 10 keyspace in your account.
        """
        print("Let's create a keyspace.")
        ks_name = q.ask(
            "Enter a name for your new keyspace.\nThe name can contain only
letters, "
            "numbers and underscores: ",
            q.non_empty,
        )
        if self.ks_wrapper.exists_keyspace(ks_name):
            print(f"A keyspace named {ks_name} exists.")
        else:
            ks_arn = self.ks_wrapper.create_keyspace(ks_name)
            ks_exists = False
            while not ks_exists:
                wait(3)
                ks_exists = self.ks_wrapper.exists_keyspace(ks_name)
```

```

        print(f"Created a new keyspace.\n\t{ks_arn}.")
    print("The first 10 keyspaces in your account are:\n")
    self.ks_wrapper.list_keyspaces(10)

@demo_func
def create_table(self):
    """
    1. Creates a table in the keyspace. The table is configured with a schema
to hold
        movie data and has point-in-time recovery enabled.
    2. Waits for the table to be in an active state.
    3. Displays schema information for the table.
    4. Lists tables in the keyspace.
    """
    print("Let's create a table for movies in your keyspace.")
    table_name = q.ask("Enter a name for your table: ", q.non_empty)
    table = self.ks_wrapper.get_table(table_name)
    if table is not None:
        print(
            f"A table named {table_name} already exists in keyspace "
            f"{self.ks_wrapper.ks_name}."
        )
    else:
        table_arn = self.ks_wrapper.create_table(table_name)
        print(f"Created table {table_name}:\n\t{table_arn}")
        table = {"status": None}
        print("Waiting for your table to be ready...")
        while table["status"] != "ACTIVE":
            wait(5)
            table = self.ks_wrapper.get_table(table_name)
        print(f"Your table is {table['status']}. Its schema is:")
        pp(table["schemaDefinition"])
        print("\nThe tables in your keyspace are:\n")
        self.ks_wrapper.list_tables()

@demo_func
def ensure_tls_cert(self):
    """
    Ensures you have a TLS certificate available to use to secure the
connection
        to the keyspace. This function downloads a default certificate or lets
you
        specify your own.
    """

```

```

print("To connect to your keyspace, you must have a TLS certificate.")
print("Checking for TLS certificate...")
cert_path = os.path.join(
    os.path.dirname(__file__), QueryManager.DEFAULT_CERT_FILE
)
if not os.path.exists(cert_path):
    cert_choice = q.ask(
        f"Press enter to download a certificate from
{QueryManager.CERT_URL} "
        f"or enter the full path to the certificate you want to use: "
    )
    if cert_choice:
        cert_path = cert_choice
    else:
        cert = requests.get(QueryManager.CERT_URL).text
        with open(cert_path, "w") as cert_file:
            cert_file.write(cert)
else:
    q.ask(f"Certificate {cert_path} found. Press Enter to continue.")
print(
    f"Certificate {cert_path} will be used to secure the connection to
your keyspace."
)
return cert_path

@demo_func
def query_table(self, qm, movie_file):
    """
    1. Adds movies to the table from a sample movie data file.
    2. Gets a list of movies from the table and lets you select one.
    3. Displays more information about the selected movie.
    """
    qm.add_movies(self.ks_wrapper.table_name, movie_file)
    movies = qm.get_movies(self.ks_wrapper.table_name)
    print(f"Added {len(movies)} movies to the table:")
    sel = q.choose("Pick one to learn more about it: ", [m.title for m in
movies])
    movie_choice = qm.get_movie(
        self.ks_wrapper.table_name, movies[sel].title, movies[sel].year
    )
    print(movie_choice.title)
    print(f"\tReleased: {movie_choice.release_date}")
    print(f"\tPlot: {movie_choice.plot}")

```



```

@demo_func
def update_and_restore_table(self, qm):
    """
    1. Updates the table by adding a column to track watched movies.
    2. Marks some of the movies as watched.
    3. Gets the list of watched movies from the table.
    4. Restores to a movies_restored table at a previous point in time.
    5. Gets the list of movies from the restored table.
    """
    print("Let's add a column to record which movies you've watched.")
    pre_update_timestamp = datetime.utcnow()
    print(
        f"Recorded the current UTC time of {pre_update_timestamp} so we can
restore the table later."
    )
    self.ks_wrapper.update_table()
    print("Waiting for your table to update...")
    table = {"status": "UPDATING"}
    while table["status"] != "ACTIVE":
        wait(5)
        table = self.ks_wrapper.get_table(self.ks_wrapper.table_name)
    print("Column 'watched' added to table.")
    q.ask(
        "Let's mark some of the movies as watched. Press Enter when you're
ready.\n"
    )
    movies = qm.get_movies(self.ks_wrapper.table_name)
    for movie in movies[:10]:
        qm.watched_movie(self.ks_wrapper.table_name, movie.title, movie.year)
        print(f"Marked {movie.title} as watched.")
    movies = qm.get_movies(self.ks_wrapper.table_name, watched=True)
    print("-" * 88)
    print("The watched movies in our table are:\n")
    for movie in movies:
        print(movie.title)
    print("-" * 88)
    if q.ask(
        "Do you want to restore the table to the way it was before all of
these\n"
        "updates? Keep in mind, this can take up to 20 minutes. (y/n) ",
        q.is_yesno,
    ):
        starting_table_name = self.ks_wrapper.table_name

```

```

        table_name_restored =
self.ks_wrapper.restore_table(pre_update_timestamp)
        table = {"status": "RESTORING"}
        while table["status"] != "ACTIVE":
            wait(10)
            table = self.ks_wrapper.get_table(table_name_restored)
        print(
            f"Restored {starting_table_name} to {table_name_restored} "
            f"at a point in time of {pre_update_timestamp}."
        )
        movies = qm.get_movies(table_name_restored)
        print("Now the movies in our table are:")
        for movie in movies:
            print(movie.title)

def cleanup(self, cert_path):
    """
    1. Deletes the table and waits for it to be removed.
    2. Deletes the keyspace.

    :param cert_path: The path of the TLS certificate used in the demo. If
the
                    certificate was downloaded during the demo, it is
removed.
    """
    if q.ask(
        f"Do you want to delete your {self.ks_wrapper.table_name} table and "
        f"{self.ks_wrapper.ks_name} keyspace? (y/n) ",
        q.is_yesno,
    ):
        table_name = self.ks_wrapper.table_name
        self.ks_wrapper.delete_table()
        table = self.ks_wrapper.get_table(table_name)
        print("Waiting for the table to be deleted.")
        while table is not None:
            wait(5)
            table = self.ks_wrapper.get_table(table_name)
        print("Table deleted.")
        self.ks_wrapper.delete_keyspace()
        print(
            "Keyspace deleted. If you chose to restore your table during the
"
            "demo, the original table is also deleted."
        )

```

```

        if cert_path == os.path.join(
            os.path.dirname(__file__), QueryManager.DEFAULT_CERT_FILE
        ) and os.path.exists(cert_path):
            os.remove(cert_path)
            print("Removed certificate that was downloaded for this demo.")

    def run_scenario(self):
        logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

        print("-" * 88)
        print("Welcome to the Amazon Keyspaces (for Apache Cassandra) demo.")
        print("-" * 88)

        self.create_keyspace()
        self.create_table()
        cert_file_path = self.ensure_tls_cert()
        # Use a context manager to ensure the connection to the keyspace is
        closed.
        with QueryManager(
            cert_file_path, boto3.DEFAULT_SESSION, self.ks_wrapper.ks_name
        ) as qm:
            self.query_table(qm, "../../resources/sample_files/movies.json")
            self.update_and_restore_table(qm)
        self.cleanup(cert_file_path)

        print("\nThanks for watching!")
        print("-" * 88)

if __name__ == "__main__":
    try:
        scenario = KeyspaceScenario(KeyspaceWrapper.from_client())
        scenario.run_scenario()
    except Exception:
        logging.exception("Something went wrong with the demo.")

```

Definieren Sie eine Klasse, die Schlüsselraum- und Tabellenaktionen umschließt.

```

class KeyspaceWrapper:
    """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
    actions."""

```

```
def __init__(self, keyspaces_client):
    """
    :param keyspaces_client: A Boto3 Amazon Keyspaces client.
    """
    self.keyspaces_client = keyspaces_client
    self.ks_name = None
    self.ks_arn = None
    self.table_name = None

    @classmethod
    def from_client(cls):
        keyspaces_client = boto3.client("keyspaces")
        return cls(keyspaces_client)

def create_keyspace(self, name):
    """
    Creates a keyspace.

    :param name: The name to give the keyspace.
    :return: The Amazon Resource Name (ARN) of the new keyspace.
    """
    try:
        response = self.keyspaces_client.create_keyspace(keyspaceName=name)
        self.ks_name = name
        self.ks_arn = response["resourceArn"]
    except ClientError as err:
        logger.error(
            "Couldn't create %s. Here's why: %s: %s",
            name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return self.ks_arn

def exists_keyspace(self, name):
    """
    Checks whether a keyspace exists.

    :param name: The name of the keyspace to look up.
```

```
:return: True when the keyspace exists. Otherwise, False.
"""
try:
    response = self.keyspaces_client.get_keyspace(keyspaceName=name)
    self.ks_name = response["keyspaceName"]
    self.ks_arn = response["resourceArn"]
    exists = True
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.info("Keyspace %s does not exist.", name)
        exists = False
    else:
        logger.error(
            "Couldn't verify %s exists. Here's why: %s: %s",
            name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
return exists

def list_keyspaces(self, limit):
    """
    Lists the keyspaces in your account.

    :param limit: The maximum number of keyspaces to list.
    """
    try:
        ks_paginator = self.keyspaces_client.get_paginator("list_keyspaces")
        for page in ks_paginator.paginate(PaginationConfig={"MaxItems":
limit}):
            for ks in page["keyspaces"]:
                print(ks["keyspaceName"])
                print(f"\t{ks['resourceArn']}")
    except ClientError as err:
        logger.error(
            "Couldn't list keyspaces. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
```

```
def create_table(self, table_name):
    """
    Creates a table in the keyspace.
    The table is created with a schema for storing movie data
    and has point-in-time recovery enabled.

    :param table_name: The name to give the table.
    :return: The ARN of the new table.
    """
    try:
        response = self.keyspaces_client.create_table(
            keyspaceName=self.ks_name,
            tableName=table_name,
            schemaDefinition={
                "allColumns": [
                    {"name": "title", "type": "text"},
                    {"name": "year", "type": "int"},
                    {"name": "release_date", "type": "timestamp"},
                    {"name": "plot", "type": "text"},
                ],
                "partitionKeys": [{"name": "year"}, {"name": "title"}],
            },
            pointInTimeRecovery={"status": "ENABLED"},
        )
    except ClientError as err:
        logger.error(
            "Couldn't create table %s. Here's why: %s: %s",
            table_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response["resourceArn"]

def get_table(self, table_name):
    """
    Gets data about a table in the keyspace.

    :param table_name: The name of the table to look up.
    :return: Data about the table.
    """
    try:
```

```
        response = self.keyspaces_client.get_table(
            keyspaceName=self.ks_name, tableName=table_name
        )
        self.table_name = table_name
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.info("Table %s does not exist.", table_name)
            self.table_name = None
            response = None
        else:
            logger.error(
                "Couldn't verify %s exists. Here's why: %s: %s",
                table_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
    return response

def list_tables(self):
    """
    Lists the tables in the keyspace.
    """
    try:
        table_paginator = self.keyspaces_client.get_paginator("list_tables")
        for page in table_paginator.paginate(keyspaceName=self.ks_name):
            for table in page["tables"]:
                print(table["tableName"])
                print(f"\t{table['resourceArn']}")
    except ClientError as err:
        logger.error(
            "Couldn't list tables in keyspace %s. Here's why: %s: %s",
            self.ks_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def update_table(self):
    """
    Updates the schema of the table.
```

This example updates a table of movie data by adding a new column that tracks whether the movie has been watched.

```

"""
try:
    self.keyspaces_client.update_table(
        keyspaceName=self.ks_name,
        tableName=self.table_name,
        addColumns=[{"name": "watched", "type": "boolean"}],
    )
except ClientError as err:
    logger.error(
        "Couldn't update table %s. Here's why: %s: %s",
        self.table_name,
        err.response["Error"]["Code"],
        err.response["Error"]["Message"],
    )
    raise

def restore_table(self, restore_timestamp):
    """
    Restores the table to a previous point in time. The table is restored
    to a new table in the same keyspace.

    :param restore_timestamp: The point in time to restore the table. This
time                               must be in UTC format.

    :return: The name of the restored table.
    """
    try:
        restored_table_name = f"{self.table_name}_restored"
        self.keyspaces_client.restore_table(
            sourceKeyspaceName=self.ks_name,
            sourceTableName=self.table_name,
            targetKeyspaceName=self.ks_name,
            targetTableName=restored_table_name,
            restoreTimestamp=restore_timestamp,
        )
    except ClientError as err:
        logger.error(
            "Couldn't restore table %s. Here's why: %s: %s",
            restore_timestamp,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],

```



```
        )
        raise
    else:
        return restored_table_name

def delete_table(self):
    """
    Deletes the table from the keyspace.
    """
    try:
        self.keyspaces_client.delete_table(
            keyspaceName=self.ks_name, tableName=self.table_name
        )
        self.table_name = None
    except ClientError as err:
        logger.error(
            "Couldn't delete table %s. Here's why: %s: %s",
            self.table_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise

def delete_keyspace(self):
    """
    Deletes the keyspace.
    """
    try:
        self.keyspaces_client.delete_keyspace(keyspaceName=self.ks_name)
        self.ks_name = None
    except ClientError as err:
        logger.error(
            "Couldn't delete keyspace %s. Here's why: %s: %s",
            self.ks_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
```

Definieren Sie eine Klasse, die eine TLS-Verbindung zu einem Schlüsselraum herstellt, sich mit Sigv4 authentifiziert und CQL-Abfragen an eine Tabelle im Schlüsselraum sendet.

```
class QueryManager:
    """
    Manages queries to an Amazon Keyspaces (for Apache Cassandra) keyspace.
    Queries are secured by TLS and authenticated by using the Signature V4
    (SigV4)
    AWS signing protocol. This is more secure than sending username and password
    with a plain-text authentication provider.

    This example downloads a default certificate to secure TLS, or lets you
    specify
    your own.

    This example uses a table of movie data to demonstrate basic queries.
    """

    DEFAULT_CERT_FILE = "sf-class2-root.crt"
    CERT_URL = f"https://certs.secureserver.net/repository/sf-class2-root.crt"

    def __init__(self, cert_file_path, boto_session, keyspace_name):
        """
        :param cert_file_path: The path and file name of the certificate used for
        TLS.
        :param boto_session: A Boto3 session. This is used to acquire your AWS
        credentials.
        :param keyspace_name: The name of the keyspace to connect.
        """
        self.cert_file_path = cert_file_path
        self.boto_session = boto_session
        self.ks_name = keyspace_name
        self.cluster = None
        self.session = None

    def __enter__(self):
        """
        Creates a session connection to the keyspace that is secured by TLS and
        authenticated by SigV4.
        """
        ssl_context = SSLContext(PROTOCOL_TLSv1_2)
```

```

        ssl_context.load_verify_locations(self.cert_file_path)
        ssl_context.verify_mode = CERT_REQUIRED
        auth_provider = SigV4AuthProvider(self.boto_session)
        contact_point = f"cassandra.
{self.boto_session.region_name}.amazonaws.com"
        exec_profile = ExecutionProfile(
            consistency_level=ConsistencyLevel.LOCAL_QUORUM,
            load_balancing_policy=DCAwareRoundRobinPolicy(),
        )
        self.cluster = Cluster(
            [contact_point],
            ssl_context=ssl_context,
            auth_provider=auth_provider,
            port=9142,
            execution_profiles={EXEC_PROFILE_DEFAULT: exec_profile},
            protocol_version=4,
        )
        self.cluster.__enter__()
        self.session = self.cluster.connect(self.ks_name)
        return self

def __exit__(self, *args):
    """
    Exits the cluster. This shuts down all existing session connections.
    """
    self.cluster.__exit__(*args)

def add_movies(self, table_name, movie_file_path):
    """
    Gets movies from a JSON file and adds them to a table in the keyspace.

    :param table_name: The name of the table.
    :param movie_file_path: The path and file name of a JSON file that
contains movie data.
    """
    with open(movie_file_path, "r") as movie_file:
        movies = json.loads(movie_file.read())
    stmt = self.session.prepare(
        f"INSERT INTO {table_name} (year, title, release_date, plot) VALUES
(?, ?, ?, ?);"
    )
    for movie in movies[:20]:
        self.session.execute(
            stmt,

```

```

        parameters=[
            movie["year"],
            movie["title"],
            date.fromisoformat(movie["info"]
["release_date"].partition("T")[0]),
            movie["info"]["plot"],
        ],
    )

def get_movies(self, table_name, watched=None):
    """
    Gets the title and year of the full list of movies from the table.

    :param table_name: The name of the movie table.
    :param watched: When specified, the returned list of movies is filtered
to
                    either movies that have been watched or movies that have
not
                    been watched. Otherwise, all movies are returned.
    :return: A list of movies in the table.
    """
    if watched is None:
        stmt = SimpleStatement(f"SELECT title, year from {table_name}")
        params = None
    else:
        stmt = SimpleStatement(
            f"SELECT title, year from {table_name} WHERE watched = %s ALLOW
FILTERING"
        )
        params = [watched]
    return self.session.execute(stmt, parameters=params).all()

def get_movie(self, table_name, title, year):
    """
    Gets a single movie from the table, by title and year.

    :param table_name: The name of the movie table.
    :param title: The title of the movie.
    :param year: The year of the movie's release.
    :return: The requested movie.
    """
    return self.session.execute(
        SimpleStatement(
            f"SELECT * from {table_name} WHERE title = %s AND year = %s"

```

```
        ),
        parameters=[title, year],
    ).one()

def watched_movie(self, table_name, title, year):
    """
    Updates a movie as having been watched.

    :param table_name: The name of the movie table.
    :param title: The title of the movie.
    :param year: The year of the movie's release.
    """
    self.session.execute(
        SimpleStatement(
            f"UPDATE {table_name} SET watched=true WHERE title = %s AND year
= %s"
        ),
        parameters=[title, year],
    )
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS -SDK für Python (Boto3).
 - [CreateKeyspace](#)
 - [CreateTable](#)
 - [DeleteKeyspace](#)
 - [DeleteTable](#)
 - [GetKeyspace](#)
 - [GetTable](#)
 - [ListKeyspaces](#)
 - [ListTables](#)
 - [RestoreTable](#)
 - [UpdateTable](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Amazon Keyspaces mit einem AWS SDK verwenden](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Bibliotheken und Tools von Amazon Keyspaces (für Apache Cassandra)

Dieser Abschnitt enthält Informationen zu Amazon Keyspaces-Bibliotheken (für Apache Cassandra), Codebeispiele und Tools.

Themen

- [Bibliotheken und Beispiele](#)
- [Hervorgehobene Repos für Beispiel- und Entwicklertools](#)

Bibliotheken und Beispiele

Die Open-Source-Bibliotheken und Entwicklertools von Amazon Keyspaces finden Sie in den Repos und GitHub in den [AWSAWSBeispiel-Repos](#).

Amazon Keyspaces (für Apache Cassandra) Entwickler-Toolkit

Dieses Repository bietet ein Docker-Image mit hilfreichen Entwicklertools für Amazon Keyspaces. Es enthält beispielsweise eine CQLSHRC-Datei mit bewährten Methoden, eine optionale AWS Authentifizierungserweiterung für cqlsh und Hilfstools für allgemeine Aufgaben. Das Toolkit ist für Amazon Keyspaces optimiert, funktioniert aber auch mit Apache Cassandra-Clustern.

<https://github.com/aws-samples/amazon-keyspaces-toolkit>.

Beispiele für Amazon Keyspaces (für Apache Cassandra)

Dieses Repo ist unsere offizielle Liste mit Amazon Keyspaces-Beispielcodes. Das Repo ist nach Sprachen in Abschnitte unterteilt (siehe [Beispiele](#)). Jede Sprache hat ihren eigenen Unterabschnitt mit Beispielen. Diese Beispiele veranschaulichen gängige Amazon Keyspaces-Servimplementierungen und -muster, die Sie beim Erstellen von Anwendungen verwenden können.

<https://github.com/aws-samples/amazon-keyspaces-examples/>.

AWSSignature Version 4 (SigV4) Authentifizierungs-Plugins

Die Plugins ermöglichen es Ihnen, den Zugriff auf Amazon Keyspaces mithilfe von AWS Identity and Access Management (IAM-) Benutzern und Rollen zu verwalten.

Java: <https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin>.

Node.js: <https://github.com/aws/aws-sigv4-auth-cassandra-nodejs-driver-plugin>.

Python: <https://github.com/aws/aws-sigv4-auth-cassandra-python-driver-plugin>.

Geh: <https://github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin>.

Hervorgehobene Repos für Beispiel- und Entwicklertools

Im Folgenden finden Sie eine Auswahl hilfreicher Community-Tools für Amazon Keyspaces (für Apache Cassandra).

Amazon Keyspaces-Protokollpuffer

Sie können Protocol Buffers (Protobuf) mit Amazon Keyspaces verwenden, um eine Alternative zu Apache Cassandra User Defined Types (UDTs) bereitzustellen. Protobuf ist ein kostenloses und plattformübergreifendes Open-Source-Datenformat, das zur Serialisierung strukturierter Daten verwendet wird. Sie können Protobuf-Daten mithilfe des BLOB CQL-Datentyps speichern und UDTs umgestalten, während strukturierte Daten in allen Anwendungen und Programmiersprachen erhalten bleiben.

Dieses Repository bietet ein Codebeispiel, das eine Verbindung zu Amazon Keyspaces herstellt, eine neue Tabelle erstellt und eine Zeile einfügt, die eine Protobuf-Nachricht enthält. Dann wird die Zeile mit starker Konsistenz gelesen.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/protobuf-user-defined-types>

AWS CloudFormation Vorlage zur Erstellung eines CloudWatch Amazon-Dashboards für Amazon Keyspaces-Metriken (für Apache Cassandra)

Dieses Repository enthält AWS CloudFormation Vorlagen zur schnellen Einrichtung von CloudWatch Metriken für Amazon Keyspaces. Die Verwendung dieser Vorlage erleichtert Ihnen den Einstieg, indem Sie bereitstellbare, vorgefertigte CloudWatch Dashboards mit häufig verwendeten Metriken bereitstellen.

<https://github.com/aws-samples/amazon-keyspaces-cloudwatch-cloudformation-templates>.

Verwenden von Amazon Keyspaces (für Apache Cassandra) mit AWS

Lambda

Das Repository enthält Beispiele, die zeigen, wie Sie von Lambda aus eine Verbindung zu Amazon Keyspaces herstellen können. Im Folgenden finden Sie einige Beispiele.

C#/.NET: <https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/dotnet/datastax-v3/connection-lambda>

Java: <https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/connection-lambda>.

Ein weiteres Lambda-Beispiel, das zeigt, wie Amazon Keyspaces von einem Python-Lambda aus bereitgestellt und verwendet werden, ist im folgenden Repo verfügbar.

<https://github.com/aws-samples/aws-keyspaces-lambda-python>

Amazon Keyspaces (für Apache Cassandra) mit Spring verwenden

Dies ist ein Beispiel, das Ihnen zeigt, wie Sie Amazon Keyspaces mit Spring Boot verwenden.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/spring>

Amazon Keyspaces (für Apache Cassandra) mit Scala verwenden

Dies ist ein Beispiel, das zeigt, wie Sie mithilfe des SIGv4-Authentifizierungs-Plug-ins mit Scala eine Verbindung zu Amazon Keyspaces herstellen.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/scala/datastax-v4/connection-sigv4>

Verwenden von Amazon Keyspaces (für Apache Cassandra) mit AWS Glue

Dies ist ein Beispiel, das zeigt, wie Amazon Keyspaces mit AWS Glue verwendet wird.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/scala/datastax-v4/aws-glue>

Amazon Keyspaces (für Apache Cassandra) Cassandra-Abfragesprache (CQL) zum Konverter AWS CloudFormation

Dieses Paket implementiert ein Befehlszeilentool zur Konvertierung von Apache Cassandra Query Language (CQL) -Skripten in AWS CloudFormation (CloudFormation) -Vorlagen, wodurch Amazon Keyspaces-Schemas einfach in Stacks verwaltet werden können. CloudFormation

<https://github.com/aws/amazon-keyspaces-cql-to-cfn-converter>.

Amazon Keyspaces (für Apache Cassandra) Helfer für den Apache Cassandra-Treiber für Java

Dieses Repository enthält Treiberrichtlinien, Beispiele und bewährte Methoden für die Verwendung des DataStax Java-Treibers mit Amazon Keyspaces (für Apache Cassandra).

<https://github.com/aws-samples/amazon-keyspaces-java-driver-helpers>.

Schnelle Kompressionsdemo von Amazon Keyspaces (für Apache Cassandra)

Dieses Repository zeigt, wie große Objekte komprimiert, gespeichert und lesen/geschrieben werden können, um die Leistung zu erhöhen und den Durchsatz und die Speicherkosten zu senken.

<https://github.com/aws-samples/amazon-keyspaces-compression-example>.

Amazon Keyspaces (für Apache Cassandra) und Amazon S3-Codec-Demo

Der benutzerdefinierte Amazon S3-Codec unterstützt die transparente, vom Benutzer konfigurierbare Zuordnung von UUID-Pointern zu Amazon S3-Objekten.

<https://github.com/aws-samples/amazon-keyspaces-large-object-s3-demo>.

Integration von Amazon Keyspaces mit Apache Spark

Apache Spark ist eine Open-Source-Engine für groß angelegte Datenanalysen. Mit Apache Spark können Sie Analysen von Daten, die in Amazon Keyspaces gespeichert sind, effizienter durchführen. Sie können Amazon Keyspaces auch verwenden, um Anwendungen mit konsistenten, single-digit-millisecond-Lesezugriff auf Analysedaten von Spark. Der Open-Source-Spark Cassandra Connector vereinfacht das Lesen und Schreiben von Daten zwischen Amazon Keyspaces und Spark.

Die Amazon Keyspaces-Unterstützung für den Spark Cassandra Connector optimiert die Ausführung von Cassandra-Workloads in Spark-basierten Analyse-Pipelines, indem ein vollständig verwalteter und serverloser Datenbankservice verwendet wird. Mit Amazon Keyspaces müssen Sie sich keine Sorgen machen, dass Spark um dieselben zugrunde liegenden Infrastrukturressourcen wie Ihre Tabellen konkurriert. Amazon Keyspaces-Tabellen skalieren automatisch auf der Grundlage Ihres Anwendungsdatenverkehrs nach oben und unten.

Das folgende Tutorial führt Sie durch die Schritte und bewährten Methoden, die erforderlich sind, um Daten mithilfe des Spark Cassandra Connectors in Amazon Keyspaces zu lesen und zu schreiben. Das Tutorial zeigt, wie Daten zu Amazon Keyspaces migriert werden, indem Daten aus einer Datei mit dem Spark Cassandra Connector geladen und in eine Amazon Keyspaces-Tabelle geschrieben werden. Anschließend zeigt das Tutorial, wie Sie die Daten mit dem Spark Cassandra Connector aus Amazon Keyspaces zurücklesen. Sie würden dies tun, um Cassandra-Workloads in Spark-basierten Analyse-Pipelines auszuführen.

Themen

- [Voraussetzungen für den Aufbau von Verbindungen zu Amazon Keyspaces mit dem Spark Cassandra Connector](#)
- [Schritt 1: Amazon Keyspaces für die Integration mit dem Apache Cassandra Spark Connector konfigurieren](#)
- [Schritt 2: Den Apache Cassandra Spark Connector konfigurieren](#)
- [Schritt 3: Erstellen Sie die Anwendungskonfigurationsdatei](#)
- [Schritt 4: Bereiten Sie die Quelldaten und die Zieltabelle in Amazon Keyspaces vor](#)
- [Schritt 5: Schreiben und Lesen von Amazon Keyspaces-Daten mit dem Apache Cassandra Spark Connector](#)
- [Behebung häufiger Fehler bei der Verwendung des Spark Cassandra Connectors mit Amazon Keyspaces](#)

Voraussetzungen für den Aufbau von Verbindungen zu Amazon Keyspaces mit dem Spark Cassandra Connector

Bevor Sie mit dem Spark Cassandra Connector eine Verbindung zu Amazon Keyspaces herstellen, müssen Sie sicherstellen, dass Sie Folgendes installiert haben. Die Kompatibilität von Amazon Keyspaces mit dem Spark Cassandra Connector wurde mit den folgenden empfohlenen Versionen getestet:

- Javaversion 8
 - Scala 2.12
 - Spark 3.4
 - Cassandra Connector 2.5 und höher
 - Cassandra-Treiber 4.12
1. Um Scala zu installieren, folgen Sie den Anweisungen unter <https://www.scala-lang.org/download/scala2.html>.
 2. Folgen Sie diesem Beispiel, um Spark 3.4.1 zu installieren.

```
curl -o spark-3.4.1-bin-hadoop3.tgz -k https://d1cdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz

# now to untar
tar -zxvf spark-3.4.1-bin-hadoop3.tgz

# set this variable.
export SPARK_HOME=$PWD/spark-3.4.1-bin-hadoop3
````
```

## Schritt 1: Amazon Keyspaces für die Integration mit dem Apache Cassandra Spark Connector konfigurieren

In diesem Schritt bestätigen Sie, dass der Partitionierer für Ihr Konto mit dem Apache Spark Connector kompatibel ist, und richten die erforderlichen IAM-Berechtigungen ein. Die folgenden bewährten Methoden helfen Ihnen dabei, ausreichend Lese-/Schreibkapazität für die Tabelle bereitzustellen.

1. Bestätigen Sie, dass `Murmur3Partitioner` ist der Standardpartitionierer für Ihr Konto. Dieser Partitionierer ist mit dem Spark Cassandra Connector kompatibel. Weitere Informationen zu Partitionierern und deren Änderung finden Sie unter [the section called “Mit Partitionierern arbeiten”](#).
2. Richten Sie Ihre IAM-Berechtigungen für Amazon Keyspaces mithilfe von Schnittstellen-VPC-Endpunkten mit Apache Spark ein.
  - Weisen Sie der Benutzertabelle Lese-/Schreibzugriff und den Systemtabellen Lesezugriff zu, wie im unten aufgeführten Beispiel für eine IAM-Richtlinie gezeigt.
  - Das Auffüllen der Tabelle `system.peers` mit Ihren verfügbaren Schnittstellen-VPC-Endpunkten ist für Kunden erforderlich, die über Spark auf Amazon Keyspaces zugreifen [VPC-Endpunkte](#).

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 },
 {
 "Sid": "ListVPCEndpoints",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeVpcEndpoints"
],
 "Resource": "*"
 }
]
}
```

3. Beachten Sie die folgenden bewährten Methoden, um eine ausreichende Lese-/Schreibdurchsatzkapazität für Ihre Amazon Keyspaces-Tabelle zu konfigurieren, um den Datenverkehr vom Spark Cassandra Connector zu unterstützen.
  - Beginnen Sie mit der Nutzung von On-Demand-Kapazität, um das Szenario zu testen.
  - Um die Kosten des Tabellendurchsatzes für Produktionsumgebungen zu optimieren, verwenden Sie einen Ratenbegrenzer für den Datenverkehr vom Konnektor und konfigurieren Sie Ihre Tabelle so, dass sie bereitgestellte Kapazität mit automatischer Skalierung nutzt. Weitere Informationen finden Sie unter [the section called “Verwaltung der Durchsatzkapazität mit Auto Scaling”](#).
  - Sie können einen festen Ratenbegrenzer verwenden, der im Lieferumfang des Cassandra-Treibers enthalten ist. Es gibt einige [auf Amazon Keyspaces zugeschnittene Ratenbegrenzer](#) in der [AWS Proben](#) Repo.
  - Weitere Informationen zur Kapazitätsverwaltung finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).

## Schritt 2: Den Apache Cassandra Spark Connector konfigurieren

Apache Spark ist eine Allzweck-Rechenplattform, die Sie auf verschiedene Arten konfigurieren können. Um Spark und den Spark Cassandra Connector für die Integration mit Amazon Keyspaces zu konfigurieren, empfehlen wir, dass Sie mit den im folgenden Abschnitt beschriebenen Mindestkonfigurationseinstellungen beginnen und diese später entsprechend Ihrer Arbeitslast erhöhen.

- Erstellen Sie Spark-Partitionsgrößen, die kleiner als 8 MB sind.

In Spark-Partitionen stellen einen atomaren Datenblock dar, der parallel ausgeführt werden kann. Wenn Sie mit dem Spark Cassandra Connector Daten in Amazon Keyspaces schreiben, gilt: Je kleiner die Spark-Partition, desto geringer ist die Anzahl der Datensätze, die die Aufgabe schreiben wird. Wenn bei einer Spark-Aufgabe mehrere Fehler auftreten, schlägt sie fehl, nachdem die angegebene Anzahl von Wiederholungsversuchen ausgeschöpft wurde. Um zu vermeiden, dass große Aufgaben wiederholt werden und viele Daten erneut verarbeitet werden, sollten Sie die Größe der Spark-Partition klein halten.

- Verwenden Sie eine geringe Anzahl gleichzeitiger Schreibvorgänge pro Executor mit einer großen Anzahl von Wiederholungsversuchen.

Amazon Keyspaces gibt Fehler bei unzureichender Kapazität als Betriebs-Timeouts an die Cassandra-Treiber zurück. Sie können Timeouts, die durch unzureichende Kapazität verursacht werden, nicht beheben, indem Sie die konfigurierte Timeout-Dauer ändern, da der Spark Cassandra Connector versucht, Anfragen transparent zu wiederholen, indem er `MultipleRetryPolicy`. Um sicherzustellen, dass Wiederholungen den Verbindungspool des Treibers nicht überlasten, verwenden Sie eine geringe Anzahl gleichzeitiger Schreibvorgänge pro Executor mit einer großen Anzahl von Wiederholungsversuchen. Der folgende Codeausschnitt ist ein Beispiel dafür.

```
spark.cassandra.query.retry.count = 500
spark.cassandra.output.concurrent.writes = 3
```

- Teilen Sie den Gesamtdurchsatz auf und verteilen Sie ihn auf mehrere Cassandra-Sitzungen.
- Der Cassandra Spark Connector erstellt eine Sitzung für jeden Spark-Executor. Stellen Sie sich diese Sitzung als Maßeinheit vor, um den erforderlichen Durchsatz und die Anzahl der erforderlichen Verbindungen zu bestimmen.
- Beginnen Sie bei der Definition der Anzahl der Kerne pro Executor und der Anzahl der Kerne pro Aufgabe niedrig und erhöhen Sie sie nach Bedarf.
- Stellen Sie Spark-Aufgabenfehler ein, um die Verarbeitung bei vorübergehenden Fehlern zu ermöglichen. Nachdem Sie sich mit den Verkehrsmerkmalen und -anforderungen Ihrer Anwendung vertraut gemacht haben, empfehlen wir, folgende Einstellungen vorzunehmen: `spark.task.maxFailures` auf einen begrenzten Wert.
- Die folgende Konfiguration kann beispielsweise zwei gleichzeitige Aufgaben pro Executor und Sitzung verarbeiten:

```
spark.executor.instances = configurable -> number of executors for the session.
spark.executor.cores = 2 -> Number of cores per executor.
spark.task.cpus = 1 -> Number of cores per task.
spark.task.maxFailures = -1
```

- Schalten Sie das Batching aus.
- Wir empfehlen, die Stapelverarbeitung zu deaktivieren, um die Direktzugriffsmuster zu verbessern. Der folgende Codeausschnitt ist ein Beispiel dafür.

```
spark.cassandra.output.batch.size.rows = 1 (Default = None)
spark.cassandra.output.batch.grouping.key = none (Default = Partition)
```

```
spark.cassandra.output.batch.grouping.buffer.size = 100 (Default = 1000)
```

- Set **SPARK\_LOCAL\_DIRS** auf eine schnelle, lokale Festplatte mit ausreichend Speicherplatz.
- Standardmäßig speichert Spark Kartenausgabedateien und belastbare verteilte Datensätze (RDDs) auf einem `/tmp` Ordner. Abhängig von der Konfiguration Ihres Spark-Hosts kann dies dazu führen, kein Platz mehr auf dem Gerät zu sein.
- Um das einzustellen **SPARK\_LOCAL\_DIRS** Umgebungsvariable für ein Verzeichnis namens `/example/spark-dir`, können Sie den folgenden Befehl verwenden.

```
export SPARK_LOCAL_DIRS=/example/spark-dir
```

## Schritt 3: Erstellen Sie die Anwendungskonfigurationsdatei

Um den Open-Source-Spark Cassandra Connector mit Amazon Keyspaces verwenden zu können, müssen Sie eine Anwendungskonfigurationsdatei bereitstellen, die die Einstellungen enthält, die für die Verbindung mit dem DataStax Java-Treiber. Sie können entweder dienstspezifische Anmeldeinformationen oder das SIGv4-Plugin verwenden, um eine Verbindung herzustellen.

Falls Sie dies noch nicht getan haben, müssen Sie das digitale Starfield-Zertifikat in eine TrustStore-Datei konvertieren. Sie können die detaillierten Schritte unter verfolgen [the section called “Bevor Sie beginnen”](#) aus dem Tutorial zur Java-Treiberverbindung. Notieren Sie sich den TrustStore-Dateipfad und das Passwort, da Sie diese Informationen benötigen, wenn Sie die Anwendungskonfigurationsdatei erstellen.

## Stellen Sie eine Verbindung mit der SIGv4-Authentifizierung her

Dieser Abschnitt zeigt Ihnen ein Beispiel `application.conf` Datei, die Sie verwenden können, wenn Sie eine Verbindung herstellen AWS Anmeldeinformationen und das SIGv4-Plugin. Falls Sie dies noch nicht getan haben, müssen Sie Ihre IAM-Zugriffsschlüssel (eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel) generieren und sie in Ihrer AWS Konfigurationsdatei oder als Umgebungsvariablen. Detaillierte Anweisungen finden Sie unter [the section called “Erforderliche Anmeldeinformationen für die Authentifizierung AWS”](#).

Ersetzen Sie im folgenden Beispiel den Dateipfad zu Ihrer TrustStore-Datei und ersetzen Sie das Passwort.

```
datastax-java-driver {
```



```

basic.contact-points = ["cassandra.us-east-1.amazonaws.com:9142"]
basic.load-balancing-policy {
 class = DefaultLoadBalancingPolicy
 local-datacenter = us-east-1
 slow-replica-avoidance = false
}
basic.request {
 consistency = LOCAL_QUORUM
}
advanced {
 auth-provider = {
 class = software.aws.mcs.auth.SigV4AuthProvider
 aws-region = us-east-1
 }
 ssl-engine-factory {
 class = DefaultSslEngineFactory
 truststore-path = "path_to_file/cassandra_truststore.jks"
 truststore-password = "password"
 }
 hostname-validation=false
}
advanced.connection.pool.local.size = 3
}

```

Aktualisieren und speichern Sie diese Konfigurationsdatei unter `/home/user1/application.conf`. Die folgenden Beispiele verwenden diesen Pfad.

## Stellen Sie eine Verbindung mit dienstspezifischen Anmeldeinformationen her

Dieser Abschnitt zeigt Ihnen ein Beispiel `application.conf`-Datei, die Sie verwenden können, wenn Sie eine Verbindung mit dienstspezifischen Anmeldeinformationen herstellen. Falls Sie dies noch nicht getan haben, müssen Sie dienstspezifische Anmeldeinformationen für Amazon Keyspaces generieren. Detaillierte Anweisungen finden Sie unter [the section called "Dienstspezifische Anmeldeinformationen"](#).

Ersetzen Sie im folgenden Beispiel `username` und `password` mit Ihren eigenen Anmeldeinformationen. Ersetzen Sie außerdem den Dateipfad zu Ihrer TrustStore-Datei und das Passwort.

```

datastax-java-driver {
 basic.contact-points = ["cassandra.us-east-1.amazonaws.com:9142"]
}

```

```
basic.load-balancing-policy {
 class = DefaultLoadBalancingPolicy
 local-datacenter = us-east-1
}
basic.request {
 consistency = LOCAL_QUORUM
}
advanced {
 auth-provider = {
 class = PlainTextAuthProvider
 username = "username"
 password = "password"
 aws-region = "us-east-1"
 }
 ssl-engine-factory {
 class = DefaultSslEngineFactory
 truststore-path = "path_to_file/cassandra_truststore.jks"
 truststore-password = "password"
 hostname-validation=false
 }
 metadata = {
 schema {
 token-map.enabled = true
 }
 }
}
}
```

Aktualisieren und speichern Sie diese Konfigurationsdatei unter `/home/user1/application.conf` zur Verwendung mit dem Codebeispiel.

## Stellen Sie eine Verbindung zu einem festen Tarif her

Um eine feste Rate pro Spark-Executor zu erzwingen, können Sie einen Request Throttler definieren. Dieser Anforderungsdrosseler begrenzt die Rate der Anfragen pro Sekunde. Der Spark Cassandra Connector stellt pro Executor eine Cassandra-Sitzung bereit. Mithilfe der folgenden Formel können Sie einen konsistenten Durchsatz für eine Tabelle erzielen.

```
max-request-per-second * numberOfExecutors = total throughput against a table
```

Sie können dieses Beispiel zur Anwendungskonfigurationsdatei hinzufügen, die Sie zuvor erstellt haben.

```
datastax-java-driver {
 advanced.throttler {
 class = RateLimitingRequestThrottler

 max-requests-per-second = 3000
 max-queue-size = 30000
 drain-interval = 1 millisecond
 }
}
```

## Schritt 4: Bereiten Sie die Quelldaten und die Zieltabelle in Amazon Keyspaces vor

In diesem Schritt erstellen Sie eine Quelldatei mit Beispieldaten und einer Amazon Keyspaces-Tabelle.

1. Erstellen Sie die Quelldatei. Sie können eine der folgenden Optionen wählen:
  - Für dieses Tutorial verwenden Sie eine Datei mit kommagetrennten Werten (CSV) mit dem Namen `keyspaces_sample_table.csv` als Quelldatei für die Datenmigration. Die bereitgestellte Beispieldatei enthält einige Datenzeilen für eine Tabelle mit dem Namen `book_awards`.
  - Laden Sie die CSV-Beispieldatei herunter (`keyspaces_sample_table.csv`), das in der folgenden Archivdatei enthalten ist [samplemigration.zip](#). Entpacken Sie das Archiv und notieren Sie sich den Pfad zu `keyspaces_sample_table.csv`.
  - Wenn Sie Ihrer eigenen CSV-Datei folgen möchten, um Daten in Amazon Keyspaces zu schreiben, stellen Sie sicher, dass die Daten randomisiert sind. Daten, die direkt aus einer Datenbank gelesen oder in Flatfiles exportiert werden, werden normalerweise nach Partition und Primärschlüssel sortiert. Der Import geordneter Daten in Amazon Keyspaces kann dazu führen, dass sie in kleinere Segmente von Amazon Keyspaces-Partitionen geschrieben werden, was zu einer ungleichmäßigen Verteilung des Datenverkehrs führt. Dies kann zu einer langsameren Leistung und höheren Fehlerraten führen.

Im Gegensatz dazu hilft die Randomisierung von Daten dabei, die integrierten Load-Balancing-Funktionen von Amazon Keyspaces zu nutzen, indem der Datenverkehr gleichmäßiger auf die Partitionen verteilt wird. Es gibt verschiedene Tools, die Sie für die Randomisierung von Daten verwenden können. Für ein Beispiel, das das Open-Source-Tool

verwendet [Shuf](#), siehe [the section called “Schritt 2: Bereiten Sie die Daten vor”](#) im Tutorial zur Datenmigration. Im Folgenden finden Sie ein Beispiel, das zeigt, wie Sie Daten wie folgt mischen: `DataFrame`.

```
import org.apache.spark.sql.functions.randval
shuffledDF = dataframe.orderBy(rand())
```

2. Erstellen Sie den Zielschlüsselraum und die Zieltabelle in Amazon Keyspaces.
  - a. Stellen Sie eine Verbindung zu Amazon Keyspaces her, indem Sie `cqlsh` verwenden, und ersetzen Sie den Dienstdienstpunkt, den Benutzernamen und das Passwort im folgenden Beispiel durch Ihre eigenen Werte.

```
cqlsh cassandra.us-east-2.amazonaws.com 9142 -u "111122223333" -
p "wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY" --ssl
```

- b. Erstellen Sie einen neuen Schlüsselraum mit dem Namen `catalog` wie im folgenden Beispiel gezeigt.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

- c. Nachdem der neue Schlüsselraum den Status `Verfügbar` hat, verwenden Sie den folgenden Code, um die Zieltabelle zu erstellen `book_awards`. Weitere Informationen zur asynchronen Ressourcenerstellung und wie Sie überprüfen können, ob eine Ressource verfügbar ist, finden Sie unter [the section called “Erstellen von Keyspaces”](#).

```
CREATE TABLE catalog.book_awards (
 year int,
 award text,
 rank int,
 category text,
 book_title text,
 author text,
 publisher text,
 PRIMARY KEY ((year, award), category, rank)
);
```

## Schritt 5: Schreiben und Lesen von Amazon Keyspaces-Daten mit dem Apache Cassandra Spark Connector

In diesem Schritt laden Sie zunächst die Daten aus der Beispieldatei in eine `DataFrame` mit dem Spark Cassandra Connector. Als Nächstes schreiben Sie die Daten aus dem `DataFrame` in Ihre Amazon Keyspaces-Tabelle. Sie können diesen Teil auch unabhängig verwenden, um beispielsweise Daten in eine Amazon Keyspaces-Tabelle zu migrieren. Schließlich lesen Sie die Daten aus Ihrer Tabelle in eine `DataFrame` mit dem Spark Cassandra Connector. Sie können diesen Teil auch unabhängig verwenden, um beispielsweise Daten aus einer Amazon Keyspaces-Tabelle zu lesen, um Datenanalysen mit Apache Spark durchzuführen.

1. Starten Sie die Spark Shell, wie im folgenden Beispiel gezeigt. Beachten Sie, dass dieses Beispiel die SIGv4-Authentifizierung verwendet.

```
./spark-shell --files application.conf --conf
spark.cassandra.connection.config.profile.path=application.conf
--packages software.aws.mcs:aws-sigv4-auth-cassandra-java-driver-
plugin:4.0.5,com.datastax.spark:spark-cassandra-connector_2.12:3.1.0 --conf
spark.sql.extensions=com.datastax.spark.connector.CassandraSparkExtensions
```

2. Importieren Sie den Spark Cassandra Connector mit dem folgenden Code.

```
import org.apache.spark.sql.cassandra._
```

3. Um Daten aus der CSV-Datei zu lesen und sie in einer `DataFrame`, können Sie das folgende Codebeispiel verwenden.

```
var df =
 spark.read.option("header","true").option("inferSchema","true").csv("keyspaces_sample_tabl
```

Sie können das Ergebnis mit dem folgenden Befehl anzeigen.

```
scala> df.show();
```

Die Ausgabe sollte in etwa so aussehen.

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
```

```

| award|year| category|rank| author| book_title|
|publisher|
+-----+-----+-----+-----+-----+-----+
+-----+
|Kwesi Manu Prize|2020| Fiction| 1| Akua Mansa| Where did you go?|
SomePublisher|
|Kwesi Manu Prize|2020| Fiction| 2| John Stiles| Yesterday|
Example Books|
|Kwesi Manu Prize|2020| Fiction| 3| Nikki Wolf|Moving to the Cha...|
AnyPublisher|
| Wolf|2020|Non-Fiction| 1| Wang Xiulan| History of Ideas|
Example Books|
| Wolf|2020|Non-Fiction| 2|Ana Carolina Silva| Science Today|
SomePublisher|
| Wolf|2020|Non-Fiction| 3| Shirley Rodriguez|The Future of Sea...|
AnyPublisher|
| Richard Roe|2020| Fiction| 1| Alejandro Rosalez| Long Summer|
SomePublisher|
| Richard Roe|2020| Fiction| 2| Arnav Desai| The Key|
Example Books|
| Richard Roe|2020| Fiction| 3| Mateo Jackson| Inside the Whale|
AnyPublisher|
+-----+-----+-----+-----+-----+-----+
+-----+

```

Sie können das Schema der Daten in der `DataFrame` im folgenden Beispiel gezeigt.

```
scala> df.printSchema
```

Die Ausgabe sollte so aussehen.

```

root
 |-- award: string (nullable = true)
 |-- year: integer (nullable = true)
 |-- category: string (nullable = true)
 |-- rank: integer (nullable = true)
 |-- author: string (nullable = true)
 |-- book_title: string (nullable = true)
 |-- publisher: string (nullable = true)

```

- Verwenden Sie den folgenden Befehl, um die Daten in das `DataFrame` zur Amazon Keyspaces-Tabelle.

```
df.write.cassandraFormat("book_awards", "catalog").mode("APPEND").save()
```

5. Um zu bestätigen, dass die Daten gespeichert wurden, können Sie sie in einen Datenrahmen zurücklesen, wie im folgenden Beispiel gezeigt.

```
var newDf = spark.read.cassandraFormat("book_awards", "catalog").load()
```

Dann können Sie die Daten anzeigen, die jetzt im Datenrahmen enthalten sind.

```
scala> newDf.show()
```

Die Ausgabe dieses Befehls sollte so aussehen.

```
+-----+-----+-----+-----+
+----+----+
| book_title| author| award| category|
|publisher|rank|year|
+-----+-----+-----+-----+
+----+----+
| Long Summer| Alejandro Rosalez| Richard Roe| Fiction|
|SomePublisher| 1|2020|
| History of Ideas| Wang Xiulan| Wolf|Non-Fiction|Example
Books	1	2020	
Where did you go?	Akua Mansa	Kwesi Manu Prize	Fiction
SomePublisher	1	2020	
Inside the Whale	Mateo Jackson	Richard Roe	Fiction
AnyPublisher	3	2020	
Yesterday	John Stiles	Kwesi Manu Prize	Fiction
Books	2	2020	
Moving to the Cha...	Nikki Wolf	Kwesi Manu Prize	Fiction
AnyPublisher	3	2020	
The Future of Sea...	Shirley Rodriguez	Wolf	Non-Fiction
AnyPublisher	3	2020	
Science Today	Ana Carolina Silva	Wolf	Non-Fiction
SomePublisher	2	2020	
The Key	Arnav Desai	Richard Roe	Fiction
Books	2	2020	
+-----+-----+-----+-----+
+----+----+
```

# Behebung häufiger Fehler bei der Verwendung des Spark Cassandra Connectors mit Amazon Keyspaces

Wenn Sie Amazon Virtual Private Cloud verwenden und eine Verbindung zu Amazon Keyspaces herstellen, werden die häufigsten Fehler bei der Verwendung des Spark-Connectors durch die folgenden Konfigurationsprobleme verursacht.

- Dem in der VPC verwendeten IAM-Benutzer oder -Rolle fehlen die erforderlichen Berechtigungen für den Zugriff auf `system.peers`-Tabelle in Amazon Keyspaces. Weitere Informationen finden Sie unter [the section called “Auffüllen von `system.peers` Tabelleneinträgen mit VPC-Endpunktinformationen der Schnittstelle”](#).
- Dem IAM-Benutzer oder der IAM-Rolle fehlen die erforderlichen Lese-/Schreibberechtigungen für die Benutzertabelle und der Lesezugriff auf die Systemtabellen in Amazon Keyspaces. Weitere Informationen finden Sie unter [the section called “Schritt 1: Amazon Keyspaces konfigurieren”](#).
- Die Java-Treiberkonfiguration deaktiviert die Überprüfung des Hostnamens beim Erstellen der SSL/TLS-Verbindung nicht. Beispiele finden Sie unter [the section called “Schritt 2: Konfigurieren Sie den Treiber”](#).

Detaillierte Schritte zur Verbindungsbehebung finden Sie unter [the section called “Verbindungsfehler am VPC-Endpunkt”](#).

Darüber hinaus können Sie Amazon verwenden `CloudWatch`-Metriken, die Ihnen bei der Behebung von Problemen mit Ihrer Spark Cassandra Connector-Konfiguration in Amazon Keyspaces helfen. Erfahren Sie mehr über die Verwendung von Amazon Keyspaces mit `CloudWatch`, siehe [the section called “Überwachung mit CloudWatch”](#).

Im folgenden Abschnitt werden die nützlichsten Metriken beschrieben, die Sie bei der Verwendung des Spark Cassandra Connectors beobachten sollten.

## `PerConnectionRequestRateExceeded`

Amazon Keyspaces hat ein Kontingent von 3.000 Anfragen pro Sekunde und Verbindung. Jeder Spark-Executor stellt eine Verbindung mit Amazon Keyspaces her. Wenn Sie mehrere Wiederholungsversuche ausführen, kann Ihr Kontingent für die Anforderungsrate pro Verbindung erschöpft sein. Wenn Sie dieses Kontingent überschreiten, gibt Amazon Keyspaces eine `PerConnectionRequestRateExceeded`-metrisch in `CloudWatch`.



Wenn du siehst `PerConnectionRequestRateExceeded` Wenn Ereignisse zusammen mit anderen System- oder Benutzerfehlern auftreten, ist es wahrscheinlich, dass Spark mehrere Wiederholungsversuche durchführt, die die zugewiesene Anzahl von Anfragen pro Verbindung überschreiten.

Wenn du siehst `PerConnectionRequestRateExceeded` Ereignisse ohne weitere Fehler, dann müssen Sie möglicherweise die Anzahl der Verbindungen in Ihren Treibereinstellungen erhöhen, um einen höheren Durchsatz zu ermöglichen, oder Sie müssen möglicherweise die Anzahl der Executoren in Ihrem Spark-Job erhöhen.

### `StoragePartitionThroughputCapacityExceeded`

Amazon Keyspaces hat ein Kontingent von 1.000 WCUs oder WRUs pro Sekunde/3.000 RCUs oder RRUs pro Sekunde und Partition. Wenn du siehst `StoragePartitionThroughputCapacityExceeded` CloudWatch Ereignisse, dies könnte darauf hindeuten, dass die Daten bei Belastung nicht randomisiert werden. Beispiele für das Mischen von Daten finden Sie unter [the section called “Schritt 4: Bereiten Sie die Quelldaten und die Zieltabelle vor”](#).

## Häufige Fehler und Warnungen

Wenn Sie Amazon Virtual Private Cloud verwenden und eine Verbindung zu Amazon Keyspaces herstellen, gibt der Cassandra-Treiber möglicherweise eine Warnmeldung über den Kontrollknoten selbst in `dersystem.peerstabelle`. Weitere Informationen finden Sie unter [the section called “Häufige Fehler und Warnungen”](#). Sie können diese Warnung getrost ignorieren.

# Fehlerbehebung bei Amazon Keyspaces (für Apache Cassandra)

Die folgenden Abschnitte enthalten Informationen zur Behebung häufiger Konfigurationsprobleme, die bei der Verwendung von Amazon Keyspaces (für Apache Cassandra) auftreten können.

Anleitungen zur Fehlerbehebung speziell für den IAM-Zugriff finden Sie unter [the section called “Fehlerbehebung”](#)

Weitere Informationen zu bewährten Sicherheitsmethoden finden Sie unter [the section called “Bewährte Methoden für die Gewährleistung der Sicherheit”](#).

## Themen

- [Behebung allgemeiner Fehler in Amazon Keyspaces](#)
- [Fehlerbehebung bei Verbindungen in Amazon Keyspaces](#)
- [Fehlerbehebung bei der Kapazitätsverwaltung in Amazon Keyspaces](#)
- [Fehlerbehebung bei der Datendefinitionssprache in Amazon Keyspaces](#)

## Behebung allgemeiner Fehler in Amazon Keyspaces

Erhalten Sie allgemeine Fehler? Im Folgenden finden Sie einige häufig auftretende Probleme und wie Sie sie lösen können.

### Allgemeine Fehler

Sie erhalten eine der folgenden Ausnahmen auf oberster Ebene, die aus vielen verschiedenen Gründen auftreten können.

- `NoNodeAvailableException`
- `NoHostAvailableException`
- `AllNodesFailedException`

Diese Ausnahmen werden vom Client-Treiber generiert und können entweder beim Herstellen der Kontrollverbindung oder beim Ausführen von Lese-/Schreib-/Vorbereitungs-, Ausführungs- und Batchanforderungen auftreten.

Wenn der Fehler beim Herstellen der Kontrollverbindung auftritt, ist dies ein Zeichen dafür, dass alle in Ihrer Anwendung angegebenen Kontaktpunkte nicht erreichbar sind. Wenn der Fehler bei der Ausführung von Abfragen mit Lese-/Schreib-/Vorbereitung/Ausführung auftritt, bedeutet dies, dass alle Wiederholungsversuche für diese Anforderung erschöpft sind. Jeder Wiederholungsversuch wird auf einem anderen Knoten versucht, wenn Sie die standardmäßige Wiederholungsrichtlinie verwenden.

## Wie isoliert man den zugrundeliegenden Fehler von Ausnahmen für Java-Treiber der obersten Ebene

Diese allgemeinen Fehler können entweder durch Verbindungsprobleme oder beim Ausführen von Lese-/Schreib-/Vorbereitungs- und Ausführungsvorgängen verursacht werden. In verteilten Systemen ist mit vorübergehenden Ausfällen zu rechnen, die durch einen erneuten Versuch der Anfrage behoben werden sollten. Der Java-Treiber versucht es nicht automatisch erneut, wenn Verbindungsfehler auftreten. Es wird daher empfohlen, die Wiederholungsrichtlinie zu implementieren, wenn Sie die Treiberverbindung in Ihrer Anwendung herstellen. Eine ausführliche Übersicht über bewährte Verbindungsmethoden finden Sie unter [the section called "Verbindungen"](#)

Standardmäßig ist der Java-Treiber für alle Anfragen `idempotence` auf `False` gesetzt, was bedeutet, dass der Java-Treiber nicht automatisch versucht, fehlgeschlagene Anfragen zu lesen/schreiben/vorzubereiten. Um diese Einstellung einzustellen `idempotence true` und den Treiber anzuweisen, fehlgeschlagene Anfragen erneut zu versuchen, können Sie dies auf verschiedene Arten tun. Hier ist ein Beispiel, wie Sie Idempotenz programmgesteuert für eine einzelne Anfrage in Ihrer Java-Anwendung festlegen können.

```
Statement s = new SimpleStatement("SELECT * FROM my_table WHERE id = 1");
s.setIdempotent(true);
```

Oder Sie können die Standard-Idempotenz für Ihre gesamte Java-Anwendung programmgesteuert festlegen, wie im folgenden Beispiel gezeigt.

```
// Make all statements idempotent by default:
cluster.getConfiguration().getQueryOptions().setDefaultIdempotence(true);
//Set the default idempotency to true in your Cassandra configuration
basic.request.default-idempotence = true
```

Eine weitere Empfehlung besteht darin, eine Wiederholungsrichtlinie auf Anwendungsebene zu erstellen. In diesem Fall muss die Anwendung die Anfrage `catch NoNodeAvailableException`

und erneut versuchen. Wir empfehlen 10 Wiederholungen mit exponentiellem Backoff, die bei 10 ms beginnen und bis zu 100 ms dauern, wobei die Gesamtzeit für alle Wiederholungen 1 Sekunde beträgt.

[Eine weitere Option besteht darin, die Amazon Keyspaces-Richtlinie für exponentielle Wiederholungen anzuwenden, wenn die auf Github verfügbare Java-Treiberbindung hergestellt wird.](#)

Vergewissern Sie sich, dass Sie Verbindungen zu mehr als einem Knoten hergestellt haben, wenn Sie die standardmäßige Wiederholungsrichtlinie verwenden. Sie können dies mit der folgenden Abfrage in Amazon Keyspaces tun.

```
SELECT * FROM system.peers;
```

Wenn die Antwort für diese Abfrage leer ist, bedeutet dies, dass Sie mit einem einzigen Knoten für Amazon Keyspaces arbeiten. Wenn Sie die standardmäßige Wiederholungsrichtlinie verwenden, gibt es keine Wiederholungsversuche, da die Standardwiederholung immer auf einem anderen Knoten erfolgt. Weitere Informationen zum Herstellen von Verbindungen über VPC-Endpunkte finden Sie unter [the section called “VPC-Endpunktverbindungen”](#)

Ein step-by-step Tutorial, das zeigt, wie Sie mithilfe des Datastax 4.x Cassandra-Treibers eine Verbindung zu Amazon Keyspaces herstellen, finden Sie unter [the section called “Authentifizierungs-Plugin für Java 4.x”](#)

## Fehlerbehebung bei Verbindungen in Amazon Keyspaces

Haben Sie Probleme beim Herstellen einer Verbindung? Hier finden Sie einige häufig auftretende Probleme und wie Sie sie lösen können.

### Fehler beim Herstellen einer Verbindung zu einem Amazon Keyspaces-Endpunkt

Fehlgeschlagene Verbindungen und Verbindungsfehler können zu unterschiedlichen Fehlermeldungen führen. Der folgende Abschnitt behandelt die häufigsten Szenarien.

Themen

- [Ich kann mit cqlsh keine Verbindung zu Amazon Keyspaces herstellen](#)
- [Ich kann mit einem Cassandra-Client-Treiber keine Verbindung zu Amazon Keyspaces herstellen](#)

## Ich kann mit cqlsh keine Verbindung zu Amazon Keyspaces herstellen

Sie versuchen, mit cqlsh eine Verbindung zu einem Amazon Keyspaces-Endpoint herzustellen, und die Verbindung schlägt mit a fehl. **Connection error**

Wenn Sie versuchen, eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen und cqlsh nicht richtig konfiguriert wurde, schlägt die Verbindung fehl. Der folgende Abschnitt enthält Beispiele für die häufigsten Konfigurationsprobleme, die zu Verbindungsfehlern führen, wenn Sie versuchen, eine Verbindung mit cqlsh herzustellen.

### Note

Wenn Sie versuchen, von einer VPC aus eine Verbindung zu Amazon Keyspaces herzustellen, sind zusätzliche Berechtigungen erforderlich. Um eine Verbindung mithilfe von VPC-Endpoints erfolgreich zu konfigurieren, folgen Sie den Schritten in [the section called "Verbindung mit VPC-Endpoints herstellen"](#)

Sie versuchen, mit cqlsh eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber einen Verbindungsfehler. **timed out**

Dies kann der Fall sein, wenn Sie nicht den richtigen Port angegeben haben, was zu dem folgenden Fehler führt.

```
cqlsh cassandra.us-east-1.amazonaws.com 9140 -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.199': error(None,
'Tried connecting to [('3.234.248.199', 9140)]. Last error: timed out)})
```

Um dieses Problem zu beheben, stellen Sie sicher, dass Sie Port 9142 für die Verbindung verwenden.

Sie versuchen, mit cqlsh eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **Name or service not known**

Dies kann der Fall sein, wenn Sie einen Endpoint verwendet haben, der falsch geschrieben ist oder nicht existiert. Im folgenden Beispiel ist der Name des Endpunkts falsch geschrieben.

```
cqlsh cassandra.us-east-1.amazon.com 9142 -u "USERNAME" -p "PASSWORD" --ssl
Traceback (most recent call last):
```

```
File "/usr/bin/cqlsh.py", line 2458, in >module>
 main(*read_options(sys.argv[1:], os.environ))
File "/usr/bin/cqlsh.py", line 2436, in main
 encoding=options.encoding)
File "/usr/bin/cqlsh.py", line 484, in __init__
 load_balancing_policy=WhiteListRoundRobinPolicy([self.hostname]),
File "/usr/share/cassandra/lib/cassandra-driver-internal-only-3.11.0-bb96859b.zip/
cassandra-driver-3.11.0-bb96859b/cassandra/policies.py", line 417, in __init__
socket.gaierror: [Errno -2] Name or service not known
```

Um dieses Problem zu beheben, wenn Sie öffentliche Endpunkte für die Verbindung verwenden, wählen Sie einen verfügbaren Endpunkt aus und stellen Sie sicher [the section called “Service-Endpunkte”](#), dass der Name des Endpunkts keine Fehler enthält. Wenn Sie VPC-Endpunkte für die Verbindung verwenden, stellen Sie sicher, dass die VPC-Endpunktinformationen in Ihrer cqlsh-Konfiguration korrekt sind.

Sie versuchen, mit cqlsh eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **OperationTimedOut**

Amazon Keyspaces setzt voraus, dass SSL für Verbindungen aktiviert ist, um eine hohe Sicherheit zu gewährleisten. Der SSL-Parameter fehlt möglicherweise, wenn Sie die folgende Fehlermeldung erhalten.

```
cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD"
Connection error: ('Unable to connect to any servers', {'3.234.248.192':
 OperationTimedOut('errors=Timed out creating connection (5 seconds),
 last_host=None',)})
#
```

Um dieses Problem zu beheben, fügen Sie dem Befehl cqlsh connection das folgende Flag hinzu.

```
--ssl
```

Sie versuchen, mit cqlsh eine Verbindung zu Amazon Keyspaces herzustellen, und erhalten eine Fehlermeldung. **SSL transport factory requires a valid certfile to be specified**

In diesem Fall fehlt der Pfad zum SSL/TLS-Zertifikat, was zu dem folgenden Fehler führt.

```
cat .cassandra/cqlshrc
```

```
[connection]
port = 9142
factory = cqlshlib.ssl.ssl_transport_factory
#

cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Validation is enabled; SSL transport factory requires a valid certfile to be specified.
Please provide path to the certfile in [ssl] section as 'certfile' option in /
root/.cassandra/cqlshrc (or use [certfiles] section) or set SSL_CERTFILE environment
variable.
#
```

Um dieses Problem zu beheben, fügen Sie den Pfad zur Zertifikatsdatei auf Ihrem Computer hinzu.

```
certfile = path_to_file/sf-class2-root.crt
```

Sie versuchen, mit `cqlsh` eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **No such file or directory**

Dies kann der Fall sein, wenn der Pfad zur Zertifikatsdatei auf Ihrem Computer falsch ist, was zu dem folgenden Fehler führt.

```
cat .cassandra/cqlshrc
[connection]
port = 9142
factory = cqlshlib.ssl.ssl_transport_factory

[ssl]
validate = true
certfile = /root/wrong_path/sf-class2-root.crt
#

cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.192': IOError(2, 'No
such file or directory')})
#
```

Um dieses Problem zu beheben, stellen Sie sicher, dass der Pfad zur Zertifikatsdatei auf Ihrem Computer korrekt ist.

Sie versuchen, mit `cqlsh` eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **[X509] PEM lib**

Dies kann der Fall sein, wenn die SSL/TLS-Zertifikatsdatei nicht gültig `sf-class2-root.crt` ist, was zu dem folgenden Fehler führt.

```
cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.241':
 error(185090057, u"Tried connecting to [('3.234.248.241', 9142)]. Last error: [X509]
 PEM lib (_ssl.c:3063)"))
#
```

Um dieses Problem zu beheben, laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter. Speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -O
```

Sie versuchen, mit `cqlsh` eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber einen **unknown** SSL-Fehler.

Dies kann der Fall sein, wenn die SSL/TLS-Zertifikatsdatei leer `sf-class2-root.crt` ist, was zu dem folgenden Fehler führt.

```
cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.220': error(0,
 u"Tried connecting to [('3.234.248.220', 9142)]. Last error: unknown error
 (_ssl.c:3063)"))
#
```

Um dieses Problem zu beheben, laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter. Speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -O
```

Sie versuchen, mit `cqlsh` eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **SSL: CERTIFICATE\_VERIFY\_FAILED**

Dies kann der Fall sein, wenn die SSL/TLS-Zertifikatsdatei nicht verifiziert werden konnte, was zu dem folgenden Fehler führt.



```
Connection error: ('Unable to connect to any servers', {'3.234.248.223':
error(1, u"Tried connecting to [('3.234.248.223', 9142)]. Last error: [SSL:
CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:727)"))})
```

Um dieses Problem zu beheben, laden Sie die Zertifikatsdatei mit dem folgenden Befehl erneut herunter. Speichern Sie `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -0
```

Sie versuchen, mit `cqlsh` eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **Last error: timed out**

Dies kann der Fall sein, wenn Sie in Ihrer Amazon EC2-Sicherheitsgruppe keine ausgehende Regel für Amazon Keyspaces konfiguriert haben, was zu dem folgenden Fehler führt.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.206': error(None,
"Tried connecting to [('3.234.248.206', 9142)]. Last error: timed out"))})
#
```

Um zu überprüfen, ob dieses Problem durch die Konfiguration der Amazon EC2 EC2-Instance verursacht wird und nicht `cqlsh`, können Sie versuchen, eine Verbindung zu Ihrem Keyspace herzustellen AWS CLI, indem Sie beispielsweise den folgenden Befehl verwenden.

```
aws keyspaces list-tables --keyspace-name 'my_keyspace'
```

Wenn auch bei diesem Befehl ein Timeout auftritt, ist die Amazon EC2 EC2-Instance nicht korrekt konfiguriert.

Um zu bestätigen, dass Sie über ausreichende Berechtigungen für den Zugriff auf Amazon Keyspaces verfügen, können Sie die verwenden, AWS CloudShell um eine Verbindung `cqlsh` herzustellen. Wenn diese Verbindungen hergestellt werden, müssen Sie die Amazon EC2 EC2-Instance konfigurieren.

Um dieses Problem zu beheben, stellen Sie sicher, dass Ihre Amazon EC2 EC2-Instance über eine ausgehende Regel verfügt, die Datenverkehr zu Amazon Keyspaces zulässt. Ist dies nicht der Fall, müssen Sie eine neue Sicherheitsgruppe für die EC2-Instance erstellen und eine Regel hinzufügen, die ausgehenden Datenverkehr zu Amazon Keyspaces-Ressourcen zulässt. Um die

Regel für ausgehenden Datenverkehr zu Amazon Keyspaces zu aktualisieren, wählen Sie CQLSH/CASSANDRA aus dem Drop-down-Menü Typ aus.

Nachdem Sie die neue Sicherheitsgruppe mit der Regel für ausgehenden Datenverkehr erstellt haben, müssen Sie sie der Instance hinzufügen. Wählen Sie die Instance aus und klicken Sie dann auf Aktionen, dann Sicherheit und dann Sicherheitsgruppen ändern. Fügen Sie die neue Sicherheitsgruppe mit der Regel für ausgehenden Datenverkehr hinzu, stellen Sie jedoch sicher, dass die Standardgruppe ebenfalls verfügbar bleibt.

Weitere Informationen zum Anzeigen und Bearbeiten von EC2-Regeln für ausgehenden Datenverkehr finden [Sie unter Regeln zu einer Sicherheitsgruppe hinzufügen im Amazon EC2 EC2-Benutzerhandbuch](#).

Sie versuchen, mit cqlsh eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **Unauthorized**

Dies kann der Fall sein, wenn Ihnen Amazon Keyspaces-Berechtigungen in der IAM-Benutzerrichtlinie fehlen, was zu dem folgenden Fehler führt.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "testuser-at-12345678910" -p
"PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.241':
AuthenticationFailed('Failed to authenticate to 3.234.248.241: Error from server:
code=2100 [Unauthorized] message="User arn:aws:iam::12345678910:user/testuser has no
permissions."' ,)})
#
```

Um dieses Problem zu beheben, stellen Sie sicher, dass der IAM-Benutzer `testuser-at-12345678910` über Zugriffsberechtigungen für Amazon Keyspaces verfügt. Beispiele für IAM-Richtlinien, die Zugriff auf Amazon Keyspaces gewähren, finden Sie unter [the section called "Beispiele für identitätsbasierte Richtlinien"](#)

Anleitungen zur Fehlerbehebung speziell für den IAM-Zugriff finden Sie unter [the section called "Fehlerbehebung"](#)

Sie versuchen, mit cqlsh eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine Fehlermeldung. **Bad credentials**

Dies kann der Fall sein, wenn der Benutzername oder das Passwort falsch ist, was zu dem folgenden Fehler führt.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.248':
AuthenticationFailed('Failed to authenticate to 3.234.248.248: Error from server:
code=0100 [Bad credentials] message="Provided username USERNAME and/or password are
incorrect"',,))
#
```

Um dieses Problem zu beheben, stellen Sie sicher, dass der **BENUTZERNAME** und **das PASSWORT** in Ihrem Code mit dem Benutzernamen und dem Passwort übereinstimmen, die Sie bei der Generierung der [dienstspezifischen Anmeldeinformationen](#) erhalten haben.

### Important

Wenn beim Versuch, eine Verbindung mit cqlsh herzustellen, weiterhin Fehler auftreten, führen Sie den Befehl mit der `--debug` Option erneut aus und fügen Sie die detaillierte Ausgabe bei der Kontaktaufnahme hinzu. [AWS Support](#)

## Ich kann mit einem Cassandra-Client-Treiber keine Verbindung zu Amazon Keyspaces herstellen

In den folgenden Abschnitten werden die häufigsten Fehler beim Herstellen einer Verbindung mit einem Cassandra-Client-Treiber beschrieben.

Sie versuchen, mithilfe des DataStax Java-Treibers eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen, erhalten aber eine **NodeUnavailableException** Fehlermeldung.

Wenn die Verbindung, über die die Anfrage versucht wird, unterbrochen ist, führt dies zu dem folgenden Fehler.

```
[com.datastax.oss.driver.api.core.NodeUnavailableException: No connection
was available to Node(endpoint=vpce-22ff22f2f22222fff-aa1bb234.cassandra.us-
west-2.vpce.amazonaws.com/11.1.1111.222:9142, hostId=1a23456b-
c77d-8888-9d99-146cb22d6ef6, hashCode=123ca4567)]
```

Um dieses Problem zu beheben, ermitteln Sie den Heartbeat-Wert und senken Sie ihn auf 30 Sekunden, falls er höher ist.

```
advanced.heartbeat.interval = 30 seconds
```

Suchen Sie dann nach dem zugehörigen Timeout und stellen Sie sicher, dass der Wert auf mindestens 5 Sekunden eingestellt ist.

```
advanced.connection.init-query-timeout = 5 seconds
```

Sie versuchen, mithilfe eines Treibers und des Sigv4-Plug-ins eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen, erhalten aber eine **AttributeError** Fehlermeldung.

Wenn die Anmeldeinformationen nicht korrekt konfiguriert sind, führt dies zu dem folgenden Fehler.

```
cassandra.cluster.NoHostAvailable: ('Unable to connect to any servers',
{'44.234.22.154:9142': AttributeError("'NoneType' object has no attribute
'access_key'")})
```

Um dieses Problem zu beheben, stellen Sie sicher, dass Sie die Ihrem IAM-Benutzer oder Ihrer IAM-Rolle zugewiesenen Anmeldeinformationen weitergeben, wenn Sie das SigV4-Plug-In verwenden. Das SigV4-Plugin benötigt die folgenden Anmeldeinformationen.

- **AWS\_ACCESS\_KEY\_ID**— Gibt einen AWS Zugriffsschlüssel an, der einem IAM-Benutzer oder einer IAM-Rolle zugeordnet ist.
- **AWS\_SECRET\_ACCESS\_KEY**— Gibt den geheimen Schlüssel an, der dem Zugriffsschlüssel zugeordnet ist. Dies ist im Wesentlichen das "Passwort" für den Zugriffsschlüssel.

Weitere Informationen über Zugriffsschlüssel und das SigV4-Plugin finden Sie unter [the section called "IAM-Anmeldeinformationen für die Authentifizierung AWS"](#).

Sie versuchen, mithilfe eines Treibers eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen, erhalten aber eine **PartialCredentialsError** Fehlermeldung.

Wenn der **AWS\_SECRET\_ACCESS\_KEY** fehlt, kann dies zu dem folgenden Fehler führen.

```
cassandra.cluster.NoHostAvailable: ('Unable to connect to any servers',
{'44.234.22.153:9142':
PartialCredentialsError('Partial credentials found in config-file, missing:
aws_secret_access_key')})
```

Um dieses Problem zu beheben, stellen Sie sicher, dass Sie bei der Verwendung des SigV4-Plug-ins **AWS\_ACCESS\_KEY\_ID** sowohl das **AWS\_SECRET\_ACCESS\_KEY** als auch das übergeben. Weitere

Informationen zu Zugriffsschlüsseln und dem SigV4-Plugin finden Sie unter. [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#)

Sie versuchen, mithilfe eines Treibers eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen, erhalten aber eine **Invalid signature** Fehlermeldung.

Dies kann der Fall sein, wenn Sie falsche Anmeldeinformationen verwendet haben, was zu dem folgenden Fehler führt.

```
cassandra.cluster.NoHostAvailable: ('Unable to connect to any servers',
 {'44.234.22.134:9142':
 AuthenticationFailed('Failed to authenticate to 44.234.22.134:9142: Error from server:
 code=0100
 [Bad credentials] message="Authentication failure: Invalid signature"'))})
```

Um dieses Problem zu beheben, stellen Sie sicher, dass die von Ihnen übergebenen Anmeldeinformationen dem IAM-Benutzer oder der IAM-Rolle zugeordnet sind, die Sie für den Zugriff auf Amazon Keyspaces konfiguriert haben. Weitere Informationen über Zugriffsschlüssel und das SigV4-Plugin finden Sie unter. [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#)

Meine VPC-Endpointverbindung funktioniert nicht richtig

Sie versuchen, über VPC-Endpunkte eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber Tokenzuordnungsfehler oder Sie haben einen niedrigen Durchsatz.

Dies kann der Fall sein, wenn die VPC-Endpointverbindung nicht korrekt konfiguriert ist.

Überprüfen Sie die folgenden Konfigurationsdetails, um diese Probleme zu beheben. Ein step-by-step Tutorial zur Konfiguration einer Verbindung über Schnittstellen-VPC-Endpunkte für Amazon Keyspaces finden Sie unter. [the section called “Verbindung mit VPC-Endpunkten herstellen”](#)

1. Vergewissern Sie sich, dass die IAM-Entität, die für die Verbindung mit Amazon Keyspaces verwendet wird, Lese-/Schreibzugriff auf die Benutzertabelle und Lesezugriff auf die Systemtabellen hat, wie im folgenden Beispiel gezeigt.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
```

```

 "Effect": "Allow",
 "Action": [
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}

```

2. Vergewissern Sie sich, dass die für die Verbindung mit Amazon Keyspaces verwendete IAM-Entität über die erforderlichen Leseberechtigungen für den Zugriff auf die VPC-Endpointinformationen auf Ihrer Amazon EC2 EC2-Instance verfügt, wie im folgenden Beispiel gezeigt.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ListVPCEndpoints",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeVpcEndpoints"
],
 "Resource": "*"
 }
]
}

```

### Note

Die verwalteten Richtlinien `AmazonKeyspacesReadOnlyAccess_v2` `AmazonKeyspacesFullAccess` enthalten die erforderlichen Berechtigungen, damit Amazon Keyspaces auf die Amazon EC2 EC2-Instance zugreifen kann, um Informationen über verfügbare Schnittstellen-VPC-Endpunkte zu lesen.

Weitere Informationen zu VPC-Endpunkten finden Sie unter [the section called “Verwenden von Schnittstellen-VPC-Endpunkten für Amazon Keyspaces”](#)

3. Stellen Sie sicher, dass die SSL-Konfiguration des Java-Treibers die Hostnamvalidierung auf False setzt, wie in diesem Beispiel gezeigt.

```
hostname-validation = false
```

Weitere Hinweise zur Treiberkonfiguration finden Sie unter [the section called “Schritt 2: Konfigurieren Sie den Treiber”](#).

4. Um zu überprüfen, ob der VPC-Endpunkt korrekt konfiguriert wurde, können Sie die folgende Anweisung in Ihrer VPC ausführen.

#### Note

Sie können Ihre lokale Entwicklerumgebung oder den Amazon Keyspaces CQL-Editor nicht verwenden, um diese Konfiguration zu bestätigen, da sie den öffentlichen Endpunkt verwenden.

```
SELECT peer FROM system.peers;
```

Die Ausgabe sollte diesem Beispiel ähneln und je nach VPC-Setup und AWS Region zwischen 2 und 6 Knoten mit privaten IP-Adressen zurückgeben.

```
peer

192.0.2.0.15
192.0.2.0.24
192.0.2.0.13
192.0.2.0.7
192.0.2.0.8

(5 rows)
```

## Ich kann keine Verbindung herstellen mit **cassandra-stress**

Sie versuchen, mit dem **cassandra-stress** Befehl eine Verbindung zu Amazon Keyspaces herzustellen, erhalten aber eine **SSL context** Fehlermeldung.

Dies passiert, wenn Sie versuchen, eine Verbindung zu Amazon Keyspaces herzustellen, der TrustStore aber nicht richtig eingerichtet ist. Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Clients zu sichern.

In diesem Fall wird der folgende Fehler angezeigt.

```
Error creating the initializing the SSL Context
```

Um dieses Problem zu lösen, folgen Sie den Anweisungen zur Einrichtung eines TrustStores, wie in diesem Thema [the section called “Bevor Sie beginnen”](#) beschrieben.

Sobald der TrustStore eingerichtet ist, sollten Sie mit dem folgenden Befehl eine Verbindung herstellen können.

```
./cassandra-stress user profile=./profile.yaml n=100 "ops(insert=1,select=1)"
cl=LOCAL_QUORUM -node "cassandra.eu-north-1.amazonaws.com" -port native=9142
-transport ssl-alg="PKIX" truststore="./cassandra_truststore.jks" truststore-
password="trustStore_pw" -mode native cql3 user="user_name" password="password"
```

## Ich kann mit IAM-Identitäten keine Verbindung herstellen

Sie versuchen, mithilfe einer IAM-Identität eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen, erhalten aber eine **Unauthorized** Fehlermeldung.

Dies passiert, wenn Sie versuchen, mithilfe einer IAM-Identität (z. B. eines IAM-Benutzers) eine Verbindung zu einer Amazon Keyspaces-Tabelle herzustellen, ohne die Richtlinie zu implementieren und dem Benutzer zuerst die erforderlichen Berechtigungen zu erteilen.

In diesem Fall wird der folgende Fehler angezeigt.

```
Connection error: ('Unable to connect to any servers', {'3.234.248.202':
AuthenticationFailed('Failed to authenticate to 3.234.248.202:
Error from server: code=2100 [Unauthorized] message="User
arn:aws:iam::1234567890123:user/testuser has no permissions."',)})
```

Überprüfen Sie die Berechtigungen des IAM-Benutzers, um dieses Problem zu beheben. Um eine Verbindung mit einem Standardtreiber herzustellen, muss ein Benutzer mindestens SELECT Zugriff



auf die Systemtabellen haben, da die meisten Treiber die Systemschlüsselbereiche/Tabellen lesen, wenn sie die Verbindung herstellen.

Beispiele für IAM-Richtlinien, die Zugriff auf das System und Benutzertabellen von Amazon Keyspaces gewähren, finden Sie unter [the section called “Zugreifen auf Amazon Keyspaces-Tabellen”](#)

Den Abschnitt zur Fehlerbehebung speziell für IAM finden Sie unter [the section called “Fehlerbehebung”](#)

Ich versuche, Daten mit cqlsh zu importieren und die Verbindung zu meiner Amazon Keyspaces-Tabelle ist unterbrochen

Sie versuchen, Daten mit cqlsh auf Amazon Keyspaces hochzuladen, erhalten aber Verbindungsfehler.

Die Verbindung zu Amazon Keyspaces schlägt fehl, nachdem der cqlsh-Client drei aufeinanderfolgende Fehler beliebiger Art vom Server empfangen hat. Der Cqlsh-Client schlägt mit der folgenden Meldung fehl.

```
Failed to import 1 rows: NoHostAvailable - , will retry later, attempt 3 of 100
```

Um diesen Fehler zu beheben, müssen Sie sicherstellen, dass die zu importierenden Daten dem Tabellenschema in Amazon Keyspaces entsprechen. Überprüfen Sie die Importdatei auf Analysefehler. Sie können versuchen, eine einzelne Datenzeile zu verwenden, indem Sie eine INSERT-Anweisung verwenden, um den Fehler zu isolieren.

Der Client versucht automatisch, die Verbindung wiederherzustellen.

## Fehlerbehebung bei der Kapazitätsverwaltung in Amazon Keyspaces

Haben Sie Probleme mit serverloser Kapazität? Im Folgenden finden Sie einige häufig auftretende Probleme und wie Sie sie lösen können.

### Kapazitätsfehler bei serverlosen Servern

In diesem Abschnitt wird beschrieben, wie Fehler im Zusammenhang mit der serverlosen Kapazitätsverwaltung erkannt und behoben werden können. Beispielsweise könnten Sie

Ereignisse mit unzureichender Kapazität beobachten, wenn Ihre Anwendung die bereitgestellte Durchsatzkapazität überschreitet.

Da es sich bei Apache Cassandra um clusterbasierte Software handelt, die für die Ausführung auf einer Flotte von Knoten konzipiert ist, enthält sie keine Ausnahmemeldungen, die sich auf serverlose Funktionen wie die Durchsatzkapazität beziehen. Die meisten Treiber verstehen nur die Fehlercodes, die in Apache Cassandra verfügbar sind, sodass Amazon Keyspaces dieselben Fehlercodes verwendet, um die Kompatibilität aufrechtzuerhalten.

Um Cassandra-Fehler den zugrunde liegenden Kapazitätsereignissen zuzuordnen, können Sie Amazon verwenden, CloudWatch um die relevanten Amazon Keyspaces-Metriken zu überwachen. Ereignisse mit unzureichender Kapazität, die zu Fehlern auf der Clientseite führen, können basierend auf der Ressource, die das Ereignis verursacht, in diese drei Gruppen eingeteilt werden:

- **Tabelle** — Wenn Sie für eine Tabelle den Modus Bereitgestellte Kapazität wählen und Ihre Anwendung den bereitgestellten Durchsatz überschreitet, treten möglicherweise Fehler aufgrund unzureichender Kapazität auf. Weitere Informationen finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).
- **Partition** — Es kann zu Ereignissen mit unzureichender Kapazität kommen, wenn der Datenverkehr auf einer bestimmten Partition 3.000 RCUs oder 1.000 WCUs überschreitet. Als bewährte Methode empfehlen wir, den Datenverkehr gleichmäßig auf die Partitionen zu verteilen. Weitere Informationen finden Sie unter [the section called “Datenmodellierung”](#).
- **Verbindung** — Möglicherweise ist der Durchsatz unzureichend, wenn Sie das Kontingent für die maximale Anzahl von Vorgängen pro Sekunde pro Verbindung überschreiten. Um den Durchsatz zu erhöhen, können Sie bei der Konfiguration der Verbindung mit dem Treiber die Anzahl der Standardverbindungen erhöhen.

Informationen zur Konfiguration von Verbindungen für Amazon Keyspaces finden Sie unter [the section called “Wie konfiguriert man Verbindungen”](#). Weitere Informationen zur Optimierung von Verbindungen über VPC-Endpunkte finden Sie unter [the section called “VPC-Endpunktverbindungen”](#)

Um festzustellen, welche Ressource das Ereignis mit unzureichender Kapazität verursacht, das den clientseitigen Fehler zurückgibt, können Sie das Dashboard in der Amazon Keyspaces-Konsole überprüfen. Standardmäßig bietet die Konsole im Abschnitt Kapazität und zugehörige CloudWatch Metriken auf der Registerkarte Kapazität der Tabelle eine aggregierte Ansicht der gängigsten Kapazitäts- und Datenverkehrsmetriken.

Um Ihr eigenes Dashboard mit Amazon zu erstellen CloudWatch, überprüfen Sie die folgenden Amazon Keyspaces-Metriken.

- `PerConnectionRequestRateExceeded`— Anfragen an Amazon Keyspaces, die das Kontingent für die Anforderungsrate pro Verbindung überschreiten. Jede Client-Verbindung zu Amazon Keyspaces kann bis zu 3000 CQL-Anfragen pro Sekunde unterstützen. Sie können mehr als 3000 Anfragen pro Sekunde ausführen, indem Sie mehrere Verbindungen herstellen.
- `ReadThrottleEvents`— Anfragen an Amazon Keyspaces, die die Lesekapazität für eine Tabelle überschreiten.
- `StoragePartitionThroughputCapacityExceeded`— Anfragen an eine Amazon Keyspaces-Speicherpartition, die die Durchsatzkapazität der Partition überschreiten. Amazon Keyspaces-Speicherpartitionen können bis zu 1000 WCU/WRU pro Sekunde und 3000 RCU/RRU pro Sekunde unterstützen. Um diese Ausnahmen zu vermeiden, empfehlen wir Ihnen, Ihr Datenmodell zu überprüfen, um den Lese-/Schreibverkehr auf mehrere Partitionen zu verteilen.
- `WriteThrottleEvents`— Anfragen an Amazon Keyspaces, die die Schreibkapazität für eine Tabelle überschreiten.

Weitere Informationen dazu finden Sie CloudWatch unter [the section called “Überwachung mit CloudWatch”](#). Eine Liste aller verfügbaren CloudWatch Metriken für Amazon Keyspaces finden Sie unter [the section called “Metriken und Dimensionen”](#).

#### Note

Um mit einem benutzerdefinierten Dashboard zu beginnen, das alle häufig beobachteten Metriken für Amazon Keyspaces anzeigt, können Sie eine vorgefertigte CloudWatch Vorlage verwenden, die GitHub im [AWS Beispiel-Repository](#) verfügbar ist.

## Themen

- [Ich erhalte von meinem Client-Treiber Fehler mit NoHostAvailable unzureichender Kapazität](#)
- [Ich erhalte beim Datenimport Schreib-Timeout-Fehler](#)
- [Ich kann die tatsächliche Speichergröße eines Schlüsselraums oder einer Tabelle nicht sehen](#)

## Ich erhalte von meinem Client-Treiber Fehler mit **NoHostAvailable** unzureichender Kapazität

Sie sehen **Read\_Timeout** unsere Ausnahmen für eine Tabelle. **Write\_Timeout**

Wiederholter Versuch, in eine Amazon Keyspaces-Tabelle mit unzureichender Kapazität zu schreiben oder aus ihr zu lesen, kann zu clientseitigen Treiberfehlern führen.

Wird verwendet CloudWatch , um Ihre bereitgestellten und tatsächlichen Durchsatzmetriken sowie Ereignisse zu unzureichender Kapazität für die Tabelle zu überwachen. Beispielsweise schlägt eine Leseanforderung, die nicht über genügend Durchsatzkapazität verfügt, mit einer Read\_Timeout Ausnahme fehl und wird der ReadThrottleEvents Metrik zugewiesen. Eine Schreibanforderung, die nicht über genügend Durchsatzkapazität verfügt, schlägt mit einer Write\_Timeout Ausnahme fehl und wird der WriteThrottleEvents Metrik zugewiesen. Weitere Informationen zu diesen Metriken finden Sie unter [the section called “Metriken und Dimensionen”](#).

Um diese Probleme zu lösen, sollten Sie eine der folgenden Optionen in Betracht ziehen.

- Erhöhen Sie den bereitgestellten Durchsatz für die Tabelle. Dies ist die maximale Menge an Durchsatzkapazität, die eine Anwendung verbrauchen kann. Weitere Informationen finden Sie unter [the section called “Lesekapazitätseinheiten und Schreibkapazitätseinheiten”](#).
- Lassen Sie den Service die Durchsatzkapazität mit automatischer Skalierung in Ihrem Namen verwalten. Weitere Informationen finden Sie unter [the section called “Verwaltung der Durchsatzkapazität mit Auto Scaling”](#).
- Wählen Sie den On-Demand-Kapazitätsmodus für die Tabelle. Weitere Informationen finden Sie unter [the section called “On-Demand-Kapazitätsmodus”](#).

Informationen dazu, wie Sie das Standardkapazitätskontingent für Ihr Konto erhöhen müssen, finden Sie unter [Kontingente](#).

Es werden Fehler im Zusammenhang mit der Überschreitung der Partitionskapazität angezeigt.

Wenn Sie den Fehler sehen, ist StoragePartitionThroughputCapacityExceeded die Partitionskapazität vorübergehend überschritten. Dies kann automatisch durch Adaptive Capacity oder On-Demand-Kapazität behoben werden. Wir empfehlen, Ihr Datenmodell zu überprüfen, um den Lese-/Schreibverkehr auf mehrere Partitionen zu verteilen, um diese Fehler zu vermeiden. Amazon Keyspaces-Speicherpartitionen können bis zu 1000 WCU/WRU pro Sekunde und 3000 RCU/RRU pro Sekunde unterstützen. Weitere Informationen darüber, wie Sie Ihr Datenmodell verbessern

können, um den Lese-/Schreibverkehr auf mehr Partitionen zu verteilen, finden Sie unter [the section called “Datenmodellierung”](#)

`Write_Timeout`-Ausnahmen können auch durch eine erhöhte Anzahl gleichzeitiger Schreibvorgänge verursacht werden, die statische und nicht statische Daten in derselben logischen Partition umfassen. Wenn erwartet wird, dass der Datenverkehr mehrere gleichzeitige Schreibvorgänge ausführt, die statische und nicht statische Daten innerhalb derselben logischen Partition umfassen, empfehlen wir, statische und nicht statische Daten getrennt zu schreiben. Das separate Schreiben der Daten trägt auch zur Optimierung der Durchsatzkosten bei.

Es werden Fehler im Zusammenhang mit der Überschreitung der Verbindungsanforderungsrate angezeigt.

Die Anzeige ist auf eine der folgenden Ursachen `PerConnectionRequestRateExceeded` zurückzuführen.

- Möglicherweise haben Sie nicht genügend Verbindungen pro Sitzung konfiguriert.
- Möglicherweise erhalten Sie weniger Verbindungen als verfügbare Peers, weil Sie die VPC-Endpunktberechtigungen nicht richtig konfiguriert haben. Weitere Informationen zu VPC-Endpunktrichtlinien finden Sie unter [the section called “Verwenden von Schnittstellen-VPC-Endpunkten für Amazon Keyspaces”](#).
- Wenn Sie einen 4.x-Treiber verwenden, überprüfen Sie, ob Sie die Hostnamenüberprüfung aktiviert haben. Der Treiber aktiviert standardmäßig die Überprüfung des TLS-Hostnamens. Diese Konfiguration führt dazu, dass Amazon Keyspaces für den Treiber als Cluster mit einem Knoten erscheint. Wir empfehlen, dass Sie die Hostnamen-Verifizierung ausschalten.

Wir empfehlen Ihnen, die folgenden bewährten Methoden zu befolgen, um sicherzustellen, dass Ihre Verbindungen und Ihr Durchsatz optimiert sind:

- Konfigurieren Sie die Durchsatzoptimierung für CQL-Abfragen.

Amazon Keyspaces unterstützt bis zu 3.000 CQL-Abfragen pro TCP-Verbindung pro Sekunde, aber die Anzahl der Verbindungen, die ein Treiber herstellen kann, ist unbegrenzt.

Die meisten Open-Source-Cassandra-Treiber richten einen Verbindungspool zu Cassandra ein und verteilen Abfragen über diesen Verbindungspool. Amazon Keyspaces stellt Treibern 9 Peer-IP-Adressen zur Verfügung. Das Standardverhalten der meisten Treiber besteht darin, eine einzige Verbindung zu jeder Peer-IP-Adresse herzustellen. Daher beträgt der maximale CQL-

Abfragedurchsatz eines Treibers, der die Standardeinstellungen verwendet, 27.000 CQL-Abfragen pro Sekunde.

Um diese Zahl zu erhöhen, empfehlen wir Ihnen, die Anzahl der Verbindungen pro IP-Adresse zu erhöhen, die Ihr Treiber in seinem Verbindungspool verwaltet. Wenn Sie beispielsweise die maximale Anzahl von Verbindungen pro IP-Adresse auf 2 festlegen, wird der maximale Durchsatz Ihres Treibers auf 54.000 CQL-Abfragen pro Sekunde verdoppelt.

- Optimieren Sie Ihre Einzelknotenverbindungen.

Standardmäßig stellen die meisten Open-Source-Cassandra-Treiber beim Einrichten einer Sitzung eine oder mehrere Verbindungen zu jeder IP-Adresse her, die in der `system.peers` Tabelle angegeben ist. Bestimmte Konfigurationen können jedoch dazu führen, dass ein Treiber eine Verbindung zu einer einzigen Amazon Keyspaces-IP-Adresse herstellt. Dies kann passieren, wenn der Treiber versucht, den SSL-Hostnamen der Peer-Knoten (z. B. DataStax Java-Treiber) zu überprüfen, oder wenn er eine Verbindung über einen VPC-Endpunkt herstellt.

Um dieselbe Verfügbarkeit und Leistung wie ein Treiber mit Verbindungen zu mehreren IP-Adressen zu erhalten, empfehlen wir Ihnen, wie folgt vorzugehen:

- Erhöhen Sie die Anzahl der Verbindungen pro IP je nach gewünschtem Client-Durchsatz auf 9 oder höher.
- Erstellen Sie eine benutzerdefinierte Wiederholungsrichtlinie, die sicherstellt, dass Wiederholungsversuche für denselben Knoten ausgeführt werden.
- Wenn Sie VPC-Endpunkte verwenden, gewähren Sie der IAM-Entität, die für die Verbindung mit Amazon Keyspaces verwendet wird, Zugriffsberechtigungen, um Ihre VPC nach den Endpunkt- und Netzwerkschnittstelleninformationen abzufragen. Dies verbessert den Lastausgleich und erhöht den Lese-/Schreibdurchsatz. Weitere Informationen finden Sie unter [???](#).

## Ich erhalte beim Datenimport Schreib-Timeout-Fehler

Beim Hochladen von Daten mithilfe des Befehls wird ein Timeout-Fehler angezeigt. **cqlsh COPY**

```
Failed to import 1 rows: WriteTimeout - Error from server: code=1100 [Coordinator node
timed out waiting for replica nodes' responses]
message="Operation timed out - received only 0 responses." info={'received_responses':
0, 'required_responses': 2, 'write_type': 'SIMPLE', 'consistency':
'LOCAL_QUORUM'}, will retry later, attempt 1 of 100
```

Amazon Keyspaces verwendet die `WriteTimeout` Ausnahmen `ReadTimeout` und, um anzuzeigen, wenn eine Schreibanforderung aufgrund unzureichender Durchsatzkapazität fehlschlägt. Um bei der Diagnose von Ausnahmen mit unzureichender Kapazität zu helfen, veröffentlicht Amazon Keyspaces die folgenden Metriken in Amazon CloudWatch.

- `WriteThrottleEvents`
- `ReadThrottledEvents`
- `StoragePartitionThroughputCapacityExceeded`

Um Fehler mit unzureichender Kapazität beim Laden von Daten zu beheben, verringern Sie die Schreibrate pro Worker oder die Gesamtaufnahmerate und versuchen Sie dann erneut, die Zeilen hochzuladen. Weitere Informationen finden Sie unter [the section called “Schritt 4: Konfigurieren von `cqlsh COPY FROM` Einstellungen”](#). [Für eine robustere Option zum Hochladen von Daten sollten Sie die Verwendung von `DSBulk` in Betracht ziehen, das im Repository verfügbar ist. GitHub](#) [step-by-step](#) Eine Anleitung dazu finden Sie unter [the section called “Daten mit `DSBulk` laden”](#).

Ich kann die tatsächliche Speichergröße eines Schlüsselraums oder einer Tabelle nicht sehen

Sie können die tatsächliche Speichergröße des Schlüsselraums oder der Tabelle nicht sehen.

Weitere Informationen zur Speichergröße Ihrer Tabelle finden Sie unter [the section called “Auswerten Ihrer Kosten auf Tabellenebene”](#). Sie können die Speichergröße auch schätzen, indem Sie mit der Berechnung der Zeilengröße in einer Tabelle beginnen. Detaillierte Anweisungen zur Berechnung der Zeilengröße finden Sie unter [the section called “Berechnung der Zeilengröße”](#).

## Fehlerbehebung bei der Datendefinitionssprache in Amazon Keyspaces

Haben Sie Probleme beim Erstellen von Ressourcen? Im Folgenden finden Sie einige häufig auftretende Probleme und wie Sie sie lösen können.

### Sprachfehler bei der Datendefinition

Amazon Keyspaces führt DDL-Operationen (Data Definition Language) asynchron aus, z. B. das Erstellen und Löschen von Schlüsselräumen und Tabellen. Wenn eine Anwendung versucht, die Ressource zu verwenden, bevor sie bereit ist, schlägt der Vorgang fehl.

Sie können den Erstellungsstatus neuer Schlüsselräume und Tabellen in der überwachen. Dies gibt an AWS Management Console, wann ein Schlüsselraum oder eine Tabelle ausstehend oder aktiv ist. Sie können den Erstellungsstatus eines neuen Schlüsselraums oder einer neuen Tabelle auch programmgesteuert überwachen, indem Sie die Systemschematabelle abfragen. Ein Schlüsselraum oder eine Tabelle wird im Systemschema sichtbar, wenn sie einsatzbereit ist.

### Note

Um die Erstellung von Schlüsselräumen mithilfe von zu optimieren AWS CloudFormation, können Sie dieses Hilfsprogramm verwenden, um CQL-Skripten in Vorlagen zu konvertieren. CloudFormation [Das Tool ist im Repository verfügbar. GitHub](#)

## Themen

- [Ich habe einen neuen Schlüsselraum erstellt, kann ihn aber nicht anzeigen oder darauf zugreifen](#)
- [Ich habe eine neue Tabelle erstellt, kann sie aber nicht anzeigen oder darauf zugreifen](#)
- [Ich versuche, eine Tabelle mithilfe von Amazon Keyspaces point-in-time Recovery \(PITR\) wiederherzustellen, aber die Wiederherstellung schlägt fehl](#)
- [Ich versuche, mit INSERT/UPDATE benutzerdefinierte TTL-Einstellungen \(Time to Live\) zu bearbeiten, aber der Vorgang schlägt fehl](#)
- [Ich versuche, Daten in meine Amazon Keyspaces-Tabelle hochzuladen, und erhalte die Fehlermeldung, dass die Anzahl der Spalten überschritten wurde](#)
- [Ich versuche, Daten in meiner Amazon Keyspaces-Tabelle zu löschen, und das Löschen schlägt für den Bereich fehl](#)

## Ich habe einen neuen Schlüsselraum erstellt, kann ihn aber nicht anzeigen oder darauf zugreifen

Sie erhalten Fehler von Ihrer Anwendung, die versucht, auf einen neuen Schlüsselraum zuzugreifen.

Wenn Sie versuchen, auf einen neu erstellten Amazon Keyspaces-Schlüsselraum zuzugreifen, der immer noch asynchron erstellt wird, erhalten Sie eine Fehlermeldung. Der folgende Fehler ist ein Beispiel.

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured keyspace mykeyspace"
```



Das empfohlene Entwurfsmuster, um zu überprüfen, wann ein neuer Keyspace einsatzbereit ist, besteht darin, die Amazon Keyspaces-Systemscematabellen (`system_schema_mcs.*`) abzufragen.

Weitere Informationen finden Sie unter [the section called “Erstellen von Keyspaces”](#).

Ich habe eine neue Tabelle erstellt, kann sie aber nicht anzeigen oder darauf zugreifen

Sie erhalten Fehler von Ihrer Anwendung, die versucht, auf eine neue Tabelle zuzugreifen.

Wenn Sie versuchen, auf eine neu erstellte Amazon Keyspaces-Tabelle zuzugreifen, die immer noch asynchron erstellt wird, erhalten Sie eine Fehlermeldung. Beispielsweise schlägt der Versuch, eine Tabelle abzufragen, die noch nicht verfügbar ist, mit einem `unconfigured table` Fehler fehl.

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured
table mykeyspace.mytable"
```

Der Versuch, die Tabelle mit `anzuzeigen`, `sync_table()` schlägt mit einem `KeyError`.

```
KeyError: 'mytable'
```

Das empfohlene Entwurfsmuster, um zu überprüfen, wann eine neue Tabelle einsatzbereit ist, besteht darin, die Amazon Keyspaces-Systemscematabellen (`system_schema_mcs.*`) abzufragen.

Dies ist die Beispielausgabe für eine Tabelle, die gerade erstellt wird.

```
user-at-123@cqlsh:system_schema_mcs> select table_name,status from
system_schema_mcs.tables where keyspace_name='example_keyspace' and
table_name='example_table';

table_name | status
-----+-----
example_table | CREATING

(1 rows)
```

Dies ist die Beispielausgabe für eine Tabelle, die aktiv ist.

```
user-at-123@cqlsh:system_schema_mcs> select table_name,status from
system_schema_mcs.tables where keyspace_name='example_keyspace' and
table_name='example_table';
```

```
table_name | status
-----+-----
example_table | ACTIVE

(1 rows)
```

Weitere Informationen finden Sie unter [the section called “Erstellen von Tabellen”](#).

## Ich versuche, eine Tabelle mithilfe von Amazon Keyspaces point-in-time Recovery (PITR) wiederherzustellen, aber die Wiederherstellung schlägt fehl

Wenn Sie versuchen, eine Amazon Keyspaces-Tabelle mit point-in-time Wiederherstellung (PITR) wiederherzustellen, und Sie sehen, dass der Wiederherstellungsvorgang beginnt, aber nicht erfolgreich abgeschlossen wird, haben Sie möglicherweise nicht alle erforderlichen Berechtigungen konfiguriert, die für den Wiederherstellungsprozess für diese bestimmte Tabelle erforderlich sind.

Zusätzlich zu den Benutzerberechtigungen benötigt Amazon Keyspaces möglicherweise Berechtigungen, um während des Wiederherstellungsvorgangs im Namen Ihres Auftraggebers Aktionen durchzuführen. Dies ist der Fall, wenn die Tabelle mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist oder wenn Sie IAM-Richtlinien verwenden, die den eingehenden Datenverkehr einschränken.

Wenn Sie beispielsweise Bedingungsschlüssel in Ihrer IAM-Richtlinie verwenden, um den Quelldatenverkehr auf bestimmte Endpunkte oder IP-Bereiche zu beschränken, schlägt der Wiederherstellungsvorgang fehl. Damit Amazon Keyspaces die Tabellenwiederherstellung im Namen Ihres Prinzipals durchführen kann, müssen Sie der IAM-Richtlinie einen `aws:ViaAWSService` globalen Bedingungsschlüssel hinzufügen.

Weitere Informationen zu Berechtigungen zum Wiederherstellen von Tabellen finden Sie unter [the section called “Berechtigungen wiederherstellen”](#)

## Ich versuche, mit INSERT/UPDATE benutzerdefinierte TTL-Einstellungen (Time to Live) zu bearbeiten, aber der Vorgang schlägt fehl

Wenn Sie versuchen, einen benutzerdefinierten TTL-Wert einzufügen oder zu aktualisieren, schlägt der Vorgang möglicherweise mit dem folgenden Fehler fehl.

```
TTL is not yet supported.
```

Um mithilfe INSERT von UPDATE OR-Operationen benutzerdefinierte TTL-Werte für Zeilen oder Spalten anzugeben, müssen Sie zuerst TTL für die Tabelle aktivieren. Sie können TTL für eine Tabelle mithilfe der `tTL` benutzerdefinierten Eigenschaft aktivieren.

Weitere Informationen zum Aktivieren benutzerdefinierter TTL-Einstellungen für Tabellen finden Sie unter [the section called “Aktivieren von für Gültigkeitsdauer \(TTL\)”](#)

Ich versuche, Daten in meine Amazon Keyspaces-Tabelle hochzuladen, und erhalte die Fehlermeldung, dass die Anzahl der Spalten überschritten wurde

Sie laden Daten hoch und haben die Anzahl der Spalten überschritten, die gleichzeitig aktualisiert werden können.

Dieser Fehler tritt auf, wenn Ihr Tabellenschema die maximale Größe von 350 KB überschreitet. Weitere Informationen finden Sie unter [Kontingente](#).

Ich versuche, Daten in meiner Amazon Keyspaces-Tabelle zu löschen, und das Löschen schlägt für den Bereich fehl

Sie versuchen, Daten anhand des Partitionsschlüssels zu löschen, und erhalten einen Fehler beim Löschen des Bereichs.

Dieser Fehler tritt auf, wenn Sie versuchen, mehr als 1.000 Zeilen in einem Löschvorgang zu löschen.

```
Range delete requests are limited by the amount of items that can be deleted in a single range.
```

Weitere Informationen finden Sie unter [the section called “Bereich löschen”](#).

Um mehr als 1.000 Zeilen innerhalb einer einzelnen Partition zu löschen, sollten Sie die folgenden Optionen in Betracht ziehen.

- Partitionsweise löschen — Wenn die Mehrheit der Partitionen weniger als 1.000 Zeilen enthält, können Sie versuchen, Daten partitionsweise zu löschen. Wenn die Partitionen mehr als 1.000 Zeilen enthalten, versuchen Sie stattdessen, sie anhand der Clusterspalte zu löschen.
- Nach Clusterspalte löschen — Wenn Ihr Modell mehrere Clusterspalten enthält, können Sie die Spaltenhierarchie verwenden, um mehrere Zeilen zu löschen. Bei Clusterspalten handelt es sich

um eine verschachtelte Struktur, und Sie können viele Zeilen löschen, indem Sie mit der Spalte der obersten Ebene arbeiten.

- Nach einzelnen Zeilen löschen — Sie können durch die Zeilen iterieren und jede Zeile anhand ihres vollständigen Primärschlüssels (Partitionsspalten und Clusterspalten) löschen.
- Als bewährte Methode sollten Sie erwägen, Ihre Zeilen auf mehrere Partitionen aufzuteilen. In Amazon Keyspaces empfehlen wir Ihnen, Ihren Durchsatz auf Tabellenpartitionen zu verteilen. Dadurch werden Daten und Zugriff gleichmäßig auf die physischen Ressourcen verteilt, wodurch der beste Durchsatz erzielt wird. Weitere Informationen finden Sie unter [the section called “Datenmodellierung”](#).

Beachten Sie auch die folgenden Empfehlungen, wenn Sie Löschvorgänge für hohe Arbeitslasten planen.

- Mit Amazon Keyspaces können Partitionen eine praktisch unbegrenzte Anzahl von Zeilen enthalten. Auf diese Weise können Sie Partitionen „breiter“ skalieren als die herkömmliche Cassandra-Richtlinie von 100 MB. Es ist nicht ungewöhnlich, dass Zeitreihen oder Ledger im Laufe der Zeit um mehr als ein Gigabyte an Daten anwachsen.
- Mit Amazon Keyspaces müssen Sie keine Verdichtungsstrategien oder Grabsteine in Betracht ziehen, wenn Sie Löschvorgänge für umfangreiche Workloads durchführen müssen. Sie können so viele Daten löschen, wie Sie möchten, ohne die Leseleistung zu beeinträchtigen.

# Serverloses Ressourcenmanagement in Amazon Keyspaces (für Apache Cassandra)

Amazon Keyspaces (für Apache Cassandra) ist serverlos. Anstatt Speicher- und Rechenressourcen für Ihre Arbeitslast über Knoten in einem Cluster bereitzustellen, zu verwalten und zu warten, weist Amazon Keyspaces Speicher- und Lese-/Schreibdurchsatzressourcen direkt Tabellen zu.

Dieses Kapitel enthält Einzelheiten zur serverlosen Ressourcenverwaltung in Amazon Keyspaces. Informationen zur Überwachung serverloser Ressourcen mit Amazon finden Sie CloudWatch unter [the section called “Überwachung mit CloudWatch”](#).

## Themen

- [Speicher in Amazon Keyspaces](#)
- [Lese-/Schreibkapazitätsmodi in Amazon Keyspaces](#)
- [Automatische Verwaltung der Durchsatzkapazität mit Amazon Keyspaces Auto Scaling](#)
- [Effektive Nutzung von Burst-Kapazität in Amazon Keyspaces](#)
- [So schätzen Sie den Kapazitätsverbrauch in Amazon Keyspaces](#)

## Speicher in Amazon Keyspaces

Amazon Keyspaces (für Apache Cassandra) stellt automatisch Speicherplatz für Tabellen bereit, basierend auf den tatsächlich in Ihrer Tabelle gespeicherten Daten. Sie müssen nicht im Voraus Speicher für Tabellen bereitstellen. Amazon Keyspaces skaliert Ihren Tabellenspeicher automatisch nach oben und unten, wenn Ihre Anwendung Daten schreibt, aktualisiert und löscht. Im Gegensatz zu herkömmlichen Apache Cassandra-Clustern benötigt Amazon Keyspaces keinen zusätzlichen Speicher, um Systemoperationen auf niedriger Ebene wie Verdichtung zu unterstützen. Sie zahlen nur für den Speicherplatz, den Sie nutzen.

Amazon Keyspaces konfiguriert Keyspaces standardmäßig mit einem Replikationsfaktor von drei. Sie können den Replikationsfaktor nicht ändern. Amazon Keyspaces repliziert Tabellendaten dreimal automatisch in mehreren AWS Availability Zones, um eine hohe Verfügbarkeit zu gewährleisten. Der Preis pro GB für Amazon Keyspaces-Speicher beinhaltet bereits die Replikation. Weitere Informationen finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\) — Preise](#).

Amazon Keyspaces überwacht kontinuierlich die Größe Ihrer Tabellen, um Ihre Speichergebühren zu ermitteln. Weitere Informationen darüber, wie Amazon Keyspaces die fakturierbare Größe der Daten berechnet, finden Sie unter. [the section called “Berechnung der Zeilengröße”](#)

## Lese-/Schreibkapazitätsmodi in Amazon Keyspaces

Amazon Keyspaces verfügt über zwei Lese-/Schreibkapazitätsmodi für die Verarbeitung von Lese- und Schreibvorgängen in Ihren Tabellen:

- Auf Abruf (Standard)
- Bereitgestellt

Der von Ihnen gewählte Lese-/Schreibkapazitätsmodus steuert, wie Ihnen der Lese- und Schreibdurchsatz in Rechnung gestellt wird und wie die Tabellendurchsatzkapazität verwaltet wird.

Themen

- [On-Demand-Kapazitätsmodus](#)
- [Bereitgestellter Durchsatzkapazitätsmodus](#)
- [Kapazitätsmodi verwalten und anzeigen](#)
- [Überlegungen beim Ändern der Kapazitätsmodi](#)

### On-Demand-Kapazitätsmodus

Der On-Demand-Kapazitätsmodus von Amazon Keyspaces (für Apache Cassandra) ist eine flexible Abrechnungsoption, mit der Tausende von Anfragen pro Sekunde ohne Kapazitätsplanung bearbeitet werden können. Diese Option bietet pay-per-request Preise für Lese- und Schreibenanfragen, sodass Sie nur für das bezahlen, was Sie tatsächlich nutzen.

Wenn Sie den On-Demand-Modus wählen, kann Amazon Keyspaces die Durchsatzkapazität für Ihre Tabelle sofort auf jedes zuvor erreichte Datenverkehrsniveau hochskalieren und dann wieder herunterfahren, wenn der Anwendungsverkehr abnimmt. Wenn der Traffic eines Workloads einen neuen Höhepunkt erreicht, passt sich der Service schnell an, um die Durchsatzkapazität für Ihre Tabelle zu erhöhen. Sie können den On-Demand-Kapazitätsmodus sowohl für neue als auch für bestehende Tabellen aktivieren.

Der On-Demand-Modus ist eine gute Option, wenn eine der folgenden Bedingungen zutrifft:

- Sie erstellen neue Tabellen mit unbekanntem Workload.
- Der Datenverkehr Ihrer Anwendung ist nicht berechenbar.
- Sie ziehen es vor, nur für Ihre tatsächliche Nutzung zu zahlen.

Um mit dem On-Demand-Modus zu beginnen, können Sie mithilfe der Konsole oder mit einigen Zeilen Cassandra Query Language (CQL) -Code eine neue Tabelle erstellen oder eine vorhandene Tabelle aktualisieren, um den On-Demand-Kapazitätsmodus zu verwenden. Weitere Informationen finden Sie unter [the section called "Tabellen"](#).

## Themen

- [Leseanforderungseinheiten und Schreibanforderungseinheiten](#)
- [Spitzenaufkommen an Datenverkehr und Skalierungseigenschaften](#)
- [Anfänglicher Durchsatz für den On-Demand-Kapazitätsmodus](#)

## Leseanforderungseinheiten und Schreibanforderungseinheiten

Bei Tabellen im On-Demand-Kapazitätsmodus müssen Sie nicht im Voraus angeben, wie viel Lese- und Schreibdurchsatz Ihre Anwendung voraussichtlich verwenden soll. Amazon Keyspaces berechnet Ihnen die Lese- und Schreibvorgänge, die Sie an Ihren Tabellen durchführen, in Form von Read Request Units (RRUs) und WRITE Request Units (WRUs).

- Eine RRU steht für eine LOCAL\_QUORUM Leseanforderung oder zwei LOCAL\_ONE Leseanforderungen für eine Zeile mit einer Größe von bis zu 4 KB. Wenn Sie eine Zeile lesen müssen, die größer als 4 KB ist, verwendet der Lesevorgang zusätzliche RRUs. Die Gesamtzahl der erforderlichen RRUs hängt von der Zeilengröße und davon ab, ob Sie Konsistenz verwenden LOCAL\_QUORUM oder LOCAL\_ONE lesen möchten. Zum Lesen einer 8-KB-Zeile sind beispielsweise 2 RRUs mit LOCAL\_QUORUM Lesekonsistenz und 1 RRU erforderlich, wenn Sie Lesekonsistenz wählen LOCAL\_ONE.
- Eine WRU steht für einen Schreibvorgang für eine Zeile mit einer Größe von bis zu 1 KB. Bei allen Schreibvorgängen wird LOCAL\_QUORUM Konsistenz verwendet, und es fallen keine zusätzlichen Gebühren für die Verwendung von Lightweight Transactions (LWTs) an. Wenn Sie eine Zeile schreiben müssen, die größer als 1 KB ist, verwendet der Schreibvorgang zusätzliche WRUs. Die Gesamtzahl der benötigten WRUs hängt von der Zeilengröße ab. Wenn Ihre Zeilengröße beispielsweise 2 KB beträgt, benötigen Sie 2 WRUs, um eine Schreibanforderung auszuführen.

Informationen zu den unterstützten Konsistenzstufen finden Sie unter [the section called “Unterstützte Cassandra-Konsistenzstufen”](#).

## Spitzenaufkommen an Datenverkehr und Skalierungseigenschaften

Amazon Keyspaces-Tabellen, die den On-Demand-Kapazitätsmodus verwenden, passen sich automatisch an das Datenverkehrsvolumen Ihrer Anwendung an. Der On-Demand-Kapazitätsmodus passt sich sofort an bis zu das Doppelte des vorherigen Höchststands des Datenverkehrs für eine Tabelle an. Beispielsweise kann das Datenverkehrsmuster Ihrer Anwendung zwischen 5.000 und 10.000 LOCAL\_QUORUM Lesevorgängen pro Sekunde variieren, wobei 10.000 Lesevorgänge pro Sekunde die vorherige Verkehrsspitze sind.

Bei diesem Muster ermöglicht der On-Demand-Kapazitätsmodus sofort einen anhaltenden Datenverkehr von bis zu 20.000 Lesevorgängen pro Sekunde. Wenn Ihre Anwendung einen Datenverkehr von 20.000 Lesevorgängen pro Sekunde aushält, wird dieser Spitzenwert zu Ihrem neuen vorherigen Spitzenwert, sodass der nachfolgende Datenverkehr bis zu 40.000 Lesevorgänge pro Sekunde erreichen kann.

Wenn Sie mehr als das Doppelte Ihres vorherigen Spitzenwerts in einer Tabelle benötigen, weist Amazon Keyspaces automatisch mehr Kapazität zu, wenn Ihr Verkehrsaufkommen steigt. Dadurch wird sichergestellt, dass Ihre Tabelle über genügend Durchsatzkapazität verfügt, um die zusätzlichen Anfragen zu verarbeiten. Sie können jedoch Fehler mit unzureichender Durchsatzkapazität feststellen, wenn Sie innerhalb von 30 Minuten das Doppelte Ihres vorherigen Spitzenwerts überschreiten.

Nehmen wir beispielsweise an, dass das Datenverkehrsmuster Ihrer Anwendung zwischen 5.000 und 10.000 stark konsistenten Lesevorgängen pro Sekunde variiert, wobei 20.000 Lesevorgänge pro Sekunde die zuvor erreichte Verkehrsspitze sind. In diesem Fall empfiehlt der Service, dass Sie Ihr Traffic-Wachstum über mindestens 30 Minuten verteilen, bevor Sie bis zu 40.000 Lesevorgänge pro Sekunde erreichen.

Informationen zur Schätzung des Lese- und Schreibkapazitätsverbrauchs einer Tabelle finden Sie unter [the section called “Schätzen Sie den Kapazitätsverbrauch”](#).

Weitere Informationen zu Standardkontingenten für Ihr Konto und deren Erhöhung finden Sie unter [Kontingente](#).



## Anfänglicher Durchsatz für den On-Demand-Kapazitätsmodus

Wenn Sie eine neue Tabelle mit aktiviertem On-Demand-Kapazitätsmodus erstellen oder eine bestehende Tabelle zum ersten Mal in den On-Demand-Kapazitätsmodus wechseln, weist die Tabelle die folgenden vorherigen Spitzeneinstellungen auf, obwohl sie zuvor keinen Datenverkehr im On-Demand-Kapazitätsmodus bedient hat:

- Neu erstellte Tabelle mit On-Demand-Kapazitätsmodus: Der bisherige Spitzenwert lag bei 2.000 WRUs und 6.000 RRUs. Sie können den vorherigen Spitzenwert sofort auf das Doppelte erhöhen. Auf diese Weise können neu erstellte On-Demand-Tabellen bis zu 4.000 WRUs und 12.000 RRUs verarbeiten.
- Bestehende Tabelle wurde in den On-Demand-Kapazitätsmodus umgeschaltet: Der vorherige Spitzenwert entspricht der Hälfte der zuvor für die Tabelle bereitgestellten WCUs und RCUs oder den Einstellungen für eine neu erstellte Tabelle mit On-Demand-Kapazitätsmodus, je nachdem, welcher Wert höher ist.

## Bereitgestellter Durchsatzkapazitätsmodus

Wenn Sie den Modus „Bereitgestellte Durchsatzkapazität“ wählen, geben Sie die Anzahl der Lese- und Schreibvorgänge pro Sekunde an, die für Ihre Anwendung erforderlich sind. Auf diese Weise können Sie Ihre Amazon Keyspaces-Nutzung so verwalten, dass sie bei oder unter einer bestimmten Anforderungsrate bleibt, um den Preis zu optimieren und die Planbarkeit aufrechtzuerhalten. Weitere Informationen zur automatischen Skalierung für den bereitgestellten Durchsatz finden Sie unter [the section called “Verwaltung der Durchsatzkapazität mit Auto Scaling”](#)

Der Kapazitätsmodus für bereitgestellten Durchsatz ist eine gute Option, wenn eine der folgenden Bedingungen zutrifft:

- Der Datenverkehr Ihrer Anwendung ist berechenbar.
- Sie führen Anwendungen aus, deren Datenverkehr konsistent ist oder schrittweise zunimmt.
- Sie können den Kapazitätsbedarf prognostizieren, um den Preis zu optimieren.

## Lesekapazitätseinheiten und Schreibkapazitätseinheiten

Für bereitgestellte Durchsatzkapazitätsmodus-Tabellen geben Sie die Durchsatzkapazität in Form von Lesekapazitätseinheiten (RCUs) und Schreibkapazitätseinheiten (WCUs) an:

- Eine RCU entspricht einem LOCAL\_QUORUM Lesevorgang pro Sekunde oder zwei LOCAL\_ONE Lesevorgängen pro Sekunde für eine Zeile mit einer Größe von bis zu 4 KB. Wenn Sie eine Zeile lesen müssen, die größer als 4 KB ist, verwendet der Lesevorgang zusätzliche RCUs.

Die Gesamtzahl der erforderlichen RCUs hängt von der Zeilengröße und davon ab, ob Sie lesen LOCAL\_QUORUM oder LOCAL\_ONE lesen möchten. Wenn Ihre Zeilengröße beispielsweise 8 KB beträgt, benötigen Sie 2 RCUs, um einen LOCAL\_QUORUM Lesevorgang pro Sekunde durchzuführen, und 1 RCU, wenn Sie Lesevorgänge wählen. LOCAL\_ONE

- Eine WCU entspricht einem Schreibvorgang pro Sekunde für eine Zeile mit einer Größe von bis zu 1 KB. Bei allen Schreibvorgängen wird LOCAL\_QUORUM Konsistenz verwendet, und es fallen keine zusätzlichen Gebühren für die Verwendung von Lightweight Transactions (LWTs) an. Wenn Sie eine Zeile schreiben müssen, die größer als 1 KB ist, verwendet der Schreibvorgang zusätzliche WCUs.

Die Gesamtzahl der benötigten WCUs hängt von der Zeilengröße ab. Wenn Ihre Zeilengröße beispielsweise 2 KB beträgt, benötigen Sie 2 WCUs, um eine Schreibanforderung pro Sekunde aufrechtzuerhalten. Weitere Hinweise zur Schätzung des Lese- und Schreibkapazitätsverbrauchs einer Tabelle finden Sie unter [the section called “Schätzen Sie den Kapazitätsverbrauch”](#).

Wenn Ihre Anwendung größere Zeilen liest oder schreibt (bis zur maximalen Zeilengröße von Amazon Keyspaces von 1 MB), verbraucht sie mehr Kapazitätseinheiten. Weitere Informationen zur Schätzung der Zeilengröße finden Sie unter [the section called “Berechnung der Zeilengröße”](#). Nehmen wir beispielsweise an, Sie erstellen eine bereitgestellte Tabelle mit 6 RCUs und 6 WCUs. Mit diesen Einstellungen hat Ihre Anwendung folgende Möglichkeiten:

- Führen Sie LOCAL\_QUORUM Lesevorgänge mit bis zu 24 KB pro Sekunde ( $4 \text{ KB} \times 6 \text{ RCUs}$ ) durch.
- Führen Sie LOCAL\_ONE Lesevorgänge mit bis zu 48 KB pro Sekunde durch (doppelt so viel Lesedurchsatz).
- Schreiben Sie bis zu 6 KB pro Sekunde ( $1 \text{ KB} \times 6 \text{ WCUs}$ ).

Der bereitgestellte Durchsatz ist die maximale Menge an Durchsatzkapazität, die eine Anwendung aus einer Tabelle verbrauchen kann. Wenn Ihre Anwendung die bereitgestellte Durchsatzkapazität überschreitet, treten möglicherweise Fehler bei unzureichender Kapazität auf.

Beispielsweise schlägt eine Leseanforderung, die nicht über genügend Durchsatzkapazität verfügt, mit einer `Read_Timeout` Ausnahme fehl und wird der `ReadThrottleEvents` Metrik zugewiesen.

Eine Schreibanforderung, die nicht über genügend Durchsatzkapazität verfügt, schlägt mit einer `Write_Timeout` Ausnahme fehl und wird der `WriteThrottleEvents` Metrik zugewiesen.

Sie können Amazon verwenden CloudWatch , um Ihre bereitgestellten und tatsächlichen Durchsatzmetriken sowie Ereignisse mit unzureichender Kapazität zu überwachen. Weitere Informationen zu diesen Metriken finden Sie unter [the section called “Metriken und Dimensionen”](#).

#### Note

Wiederholte Fehler aufgrund unzureichender Kapazität können zu clientseitigen treiberspezifischen Ausnahmen führen, z. B. schlägt der DataStax Java-Treiber mit einem fehl. `NoHostAvailableException`

Um die Durchsatzkapazitätseinstellungen für Tabellen zu ändern, können Sie die Anweisung AWS Management Console oder die `ALTER TABLE` Anweisung mithilfe von CQL verwenden. Weitere Informationen finden Sie unter [the section called “ALTER TABLE”](#)

Weitere Informationen zu Standardkontingenten für Ihr Konto und deren Erhöhung finden Sie unter [Kontingente](#).

## Kapazitätsmodi verwalten und anzeigen

Sie können die Systemtabelle im Amazon Keyspaces-Systemschlüsselraum abfragen, um Informationen zum Kapazitätsmodus einer Tabelle zu überprüfen. Sie können auch sehen, ob eine Tabelle den On-Demand-Durchsatzkapazitätsmodus oder den bereitgestellten Durchsatzkapazitätsmodus verwendet. Wenn die Tabelle mit dem Modus „Bereitgestellte Durchsatzkapazität“ konfiguriert ist, können Sie die für die Tabelle bereitgestellte Durchsatzkapazität sehen.

### Beispiel

```
SELECT * from system_schema_mcs.tables where keyspace_name = 'mykeyspace' and
table_name = 'mytable';
```

Eine Tabelle, die mit dem On-Demand-Kapazitätsmodus konfiguriert ist, gibt Folgendes zurück.

```
{
 'capacity_mode': {
```

```

 'last_update_to_pay_per_request_timestamp':
'1579551547603',
 'throughput_mode': 'PAY_PER_REQUEST'
 }
}

```

Eine mit dem Modus „Bereitgestellte Durchsatzkapazität“ konfigurierte Tabelle gibt Folgendes zurück.

```

 {
 'capacity_mode': {
 'last_update_to_pay_per_request_timestamp':
'1579048006000',
 'read_capacity_units': '5000',
 'throughput_mode': 'PROVISIONED',
 'write_capacity_units': '6000'
 }
 }
}

```

Der `last_update_to_pay_per_request_timestamp` Wert wird in Millisekunden gemessen.

Um die bereitgestellte Durchsatzkapazität für eine Tabelle zu ändern, verwenden Sie [the section called “ALTER TABLE”](#)

## Überlegungen beim Ändern der Kapazitätsmodi

Wenn Sie eine Tabelle vom Bereitstellungskapazitätsmodus in den On-Demand-Kapazitätsmodus wechseln, nimmt Amazon Keyspaces mehrere Änderungen an der Struktur Ihrer Tabelle und Partitionen vor. Dieser Vorgang kann einige Minuten dauern. Während der Umstellungsphase liefert Ihre Tabelle einen Durchsatz, der den zuvor bereitgestellten WCU- und RCU-Mengen entspricht.

Wenn Sie vom On-Demand-Kapazitätsmodus zurück in den Modus mit bereitgestellter Kapazität wechseln, liefert Ihre Tabelle einen Durchsatz, der dem vorherigen Spitzenwert entspricht, der erreicht wurde, als die Tabelle in den On-Demand-Kapazitätsmodus versetzt wurde.

### Note

Sie können den Kapazitätsmodus nur einmal innerhalb von 24 Stunden vom Bereitstellungs- zum On-Demand-Kapazitätsmodus wechseln.

# Automatische Verwaltung der Durchsatzkapazität mit Amazon Keyspaces Auto Scaling

Viele Datenbank-Workloads sind zyklischer Natur oder lassen sich nur schwer vorhersagen. Nehmen wir als Beispiel eine Social Networking-App, bei der die meisten Benutzer tagsüber aktiv sind. Die Datenbank muss in der Lage sein, die Aktivitäten am Tag zu verarbeiten, während in der Nacht nicht so viel Durchsatz erforderlich ist.

Ein weiteres Beispiel wäre eine neue Mobile Gaming-App, die sich schnell großer Beliebtheit erfreut. Wenn das Spiel sehr beliebt wird, könnte es die verfügbaren Datenbankressourcen überschreiten, was zu einer langsamen Leistung und unzufriedenen Kunden führen würde. Diese Arten von Workloads erfordern häufig manuelle Eingriffe, um die Datenbankressourcen nach oben oder unten zu skalieren, um sie an die jeweiligen Nutzungsschwankungen anzupassen.

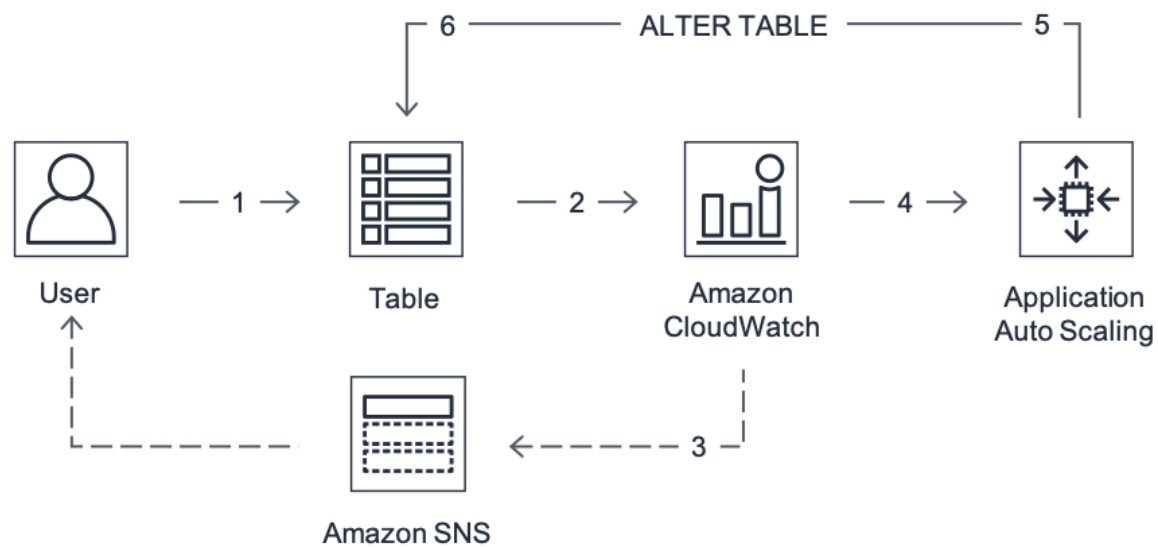
Amazon Keyspaces (für Apache Cassandra) unterstützt Sie bei der effizienten Bereitstellung von Durchsatzkapazität für variable Workloads, indem die Durchsatzkapazität automatisch an den tatsächlichen Anwendungsverkehr angepasst wird. Amazon Keyspaces verwendet den Application Auto Scaling-Service, um die Lese- und Schreibkapazität einer Tabelle in Ihrem Namen zu erhöhen oder zu verringern. Weitere Informationen zu Application Auto Scaling finden Sie im [Application Auto Scaling Scaling-Benutzerhandbuch](#).

## Note

Informationen zum schnellen Einstieg in die automatische Skalierung von Amazon Keyspaces finden Sie unter [the section called “Verwenden der Konsole”](#). Informationen zur Verwaltung von Amazon Keyspaces-Skalierungsrichtlinien mit Cassandra Query Language (CQL) finden Sie unter [the section called “Verwenden von CQL”](#). Informationen zur Verwaltung von Amazon Keyspaces-Skalierungsrichtlinien mithilfe der CLI finden Sie unter [the section called “Verwenden der -CLI”](#).

## So funktioniert die automatische Skalierung von Amazon Keyspace

Das folgende Diagramm bietet einen allgemeinen Überblick darüber, wie die automatische Skalierung von Amazon Keyspace die Durchsatzkapazität für eine Tabelle verwaltet.



Um die automatische Skalierung für eine Tabelle zu aktivieren, erstellen Sie eine Skalierungsrichtlinie. Die Skalierungsrichtlinie gibt an, ob Sie die Lese- oder Schreibkapazität (oder beides) skalieren möchten, sowie die Mindest- und Höchstwerte der bereitgestellten Kapazitätseinheiten für die Tabelle.

Die Skalierungsrichtlinie definiert auch eine Zielauslastung. Die Zielauslastung ist das Verhältnis der verbrauchten Kapazitätseinheiten zu den bereitgestellten Kapazitätseinheiten zu einem bestimmten Zeitpunkt, ausgedrückt als Prozentsatz. Die automatische Skalierung verwendet einen Algorithmus zur Zielverfolgung, um den bereitgestellten Durchsatz der Tabelle entsprechend den tatsächlichen Workloads nach oben oder unten anzupassen. Auf diese Weise wird sichergestellt, dass die tatsächliche Kapazitätsauslastung bei oder in der Nähe Ihrer Zielauslastung bleibt.

Sie können die Zielauslastung für die automatische Skalierung zwischen 20 und 90 Prozent für Ihre Lese- und Schreibkapazität festlegen. Die standardmäßige Zielauslastungsrate beträgt 70 Prozent. Sie können die Zielauslastung auf einen niedrigeren Prozentsatz festlegen, wenn sich Ihr Datenverkehr schnell ändert und Sie möchten, dass die Kapazität früher skaliert wird. Sie können die Zielauslastungsrate auch auf eine höhere Rate festlegen, wenn sich Ihr Anwendungsdatenverkehr langsamer ändert und Sie die Durchsatzkosten senken möchten.

Weitere Informationen zu Skalierungsrichtlinien finden Sie unter [Target-Tracking-Skalierungsrichtlinien für Application Auto Scaling](#) im [Application Auto Scaling Scaling-Benutzerhandbuch](#).

Wenn Sie eine Skalierungsrichtlinie erstellen, erstellt Amazon Keyspaces in Ihrem Namen zwei Paare von CloudWatch Amazon-Alarmen. Jedes Paar steht für Ihre Ober- und Untergrenzen für bereitgestellte und verbrauchte Durchsatzeinstellungen. Diese CloudWatch Alarme werden ausgelöst, wenn die tatsächliche Auslastung der Tabelle über einen längeren Zeitraum von Ihrer Zielauslastung abweicht. Weitere Informationen über Amazon CloudWatch finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Wenn einer der CloudWatch Alarme ausgelöst wird, sendet Ihnen Amazon Simple Notification Service (Amazon SNS) eine Benachrichtigung (sofern Sie ihn aktiviert haben). Der CloudWatch Alarm ruft dann Application Auto Scaling auf, um Ihre Skalierungsrichtlinie auszuwerten. Dies wiederum sendet eine Alter-Table-Anfrage an Amazon Keyspaces, um die bereitgestellte Kapazität der Tabelle nach Bedarf nach oben oder unten anzupassen. Weitere Informationen zu Amazon SNS SNS-Benachrichtigungen finden Sie unter [Amazon SNS SNS-Benachrichtigungen einrichten](#).

Amazon Keyspaces verarbeitet die Alter-Table-Anfrage, indem es die bereitgestellte Durchsatzkapazität der Tabelle erhöht (oder verringert), sodass sie sich Ihrer Zielauslastung annähert.

#### Note

Amazon Keyspaces Auto Scaling ändert die bereitgestellten Durchsatzeinstellungen nur, wenn die tatsächliche Arbeitslast über einen längeren Zeitraum von mehreren Minuten erhöht (oder reduziert) bleibt. Der Ziel-Tracking-Algorithmus versucht, dafür zu sorgen, dass die Zielauslastung langfristig Ihrem gewählten Wert (nahezu) entspricht. Plötzliche Aktivitätsspitzen werden von der integrierten Burst-Kapazität der Tabelle bewältigt.

## So funktioniert Auto Scaling für Tabellen mit mehreren Regionen

Um sicherzustellen, dass immer genügend Lese- und Schreibkapazität für alle Tabellenreplikate in einer Tabelle mit mehreren Regionen im Modus „Bereitgestellte Kapazität“ vorhanden ist, empfehlen wir Ihnen, Amazon Keyspaces Auto Scaling zu konfigurieren. AWS-Regionen

Wenn Sie eine Tabelle mit mehreren Regionen im Bereitstellungsmodus mit Auto Scaling verwenden, können Sie Auto Scaling nicht für ein einzelnes Tabellenreplikat deaktivieren. Sie können jedoch die Lese-Auto-Scaling-Einstellungen der Tabelle für verschiedene Regionen anpassen. Sie können beispielsweise für jede Region, in der die Tabelle repliziert wird, unterschiedliche Einstellungen für die Lesekapazität und das auto Skalieren von Lesevorgängen angeben.

Die Auto-Scaling-Leseinstellungen, die Sie für ein Tabellenreplikat in einer bestimmten Region konfigurieren, überschreiben die allgemeinen Auto Scaling-Einstellungen der Tabelle. Die Schreibkapazität muss jedoch über alle Tabellenreplikate hinweg synchronisiert bleiben, um sicherzustellen, dass genügend Kapazität für die Replikation von Schreibvorgängen in allen Regionen vorhanden ist.

Amazon Keyspaces Auto Scaling aktualisiert unabhängig die bereitgestellte Kapazität der Tabelle in jeder Tabelle auf der AWS-Region Grundlage der Nutzung in dieser Region. Daher kann die bereitgestellte Kapazität in jeder Region für eine Tabelle mit mehreren Regionen unterschiedlich sein, wenn Auto Scaling aktiv ist.

Sie können die Auto Scaling-Einstellungen einer Tabelle mit mehreren Regionen und ihren Replikaten mithilfe der Amazon Keyspaces-Konsole AWS CLI, API oder CQL konfigurieren. Weitere Informationen zum Erstellen und Aktualisieren von Auto Scaling-Einstellungen für Tabellen mit mehreren Regionen finden Sie unter [the section called “So verwenden Sie die Multi-Region-Replikation”](#).

#### Note

Wenn Sie Auto Scaling für Tabellen mit mehreren Regionen verwenden, müssen Sie immer Amazon Keyspaces-API-Operationen verwenden, um Auto-Scaling-Einstellungen zu konfigurieren. Wenn Sie die API-Operationen von Application Auto Scaling direkt verwenden, um Auto Scaling-Einstellungen zu konfigurieren, können Sie die AWS-Regionen Tabelle mit mehreren Regionen nicht angeben. Dies kann dazu führen, dass Konfigurationen nicht unterstützt werden.

## Nutzungshinweise

Bevor Sie mit der automatischen Skalierung von Amazon Keyspaces beginnen, sollten Sie Folgendes beachten:

- Die automatische Skalierung von Amazon Keyspace kann die Lese- oder Schreibkapazität gemäß Ihrer Skalierungsrichtlinie beliebig oft erhöhen. Alle Amazon Keyspaces-Kontingente bleiben in Kraft, wie unter beschrieben. [Kontingente](#)
- Die automatische Skalierung von Amazon Keyspace verhindert nicht, dass Sie die bereitgestellten Durchsatzeinstellungen manuell ändern. Diese manuellen Anpassungen wirken sich nicht auf bestehende CloudWatch Alarmer aus, die mit der Skalierungsrichtlinie verknüpft sind.



- Wenn Sie die Konsole verwenden, um eine Tabelle mit bereitgestellter Durchsatzkapazität zu erstellen, ist die automatische Skalierung von Amazon Keyspace standardmäßig aktiviert. Sie können Ihre Einstellungen für die automatische Skalierung jederzeit ändern. Weitere Informationen finden Sie unter [the section called “Verwenden der Konsole”](#).
- Wenn Sie Skalierungsrichtlinien erstellen, sollten Sie die Skalierungsrichtlinien AWS CloudFormation so verwalten, dass der Stack mit der Stack-Vorlage synchronisiert ist. AWS CloudFormation Wenn Sie Skalierungsrichtlinien von Amazon Keyspaces ändern, werden diese beim Zurücksetzen des Stacks mit den Originalwerten aus der AWS CloudFormation Stack-Vorlage überschrieben.
- Wenn Sie CloudTrail die automatische Skalierung von Amazon Keyspaces überwachen, werden Ihnen möglicherweise Warnmeldungen für Aufrufe angezeigt, die von Application Auto Scaling im Rahmen des Konfigurationsvalidierungsprozesses getätigt wurden. Sie können diese Benachrichtigungen herausfiltern, indem Sie das `invokedBy` Feld verwenden, das `application-autoscaling.amazonaws.com` für diese Validierungsprüfungen enthält.

## Verwalten von Auto-Scaling-Richtlinien für Amazon Keyspaces mit der Konsole

Sie können die Konsole verwenden, um die automatische Skalierung von Amazon Keyspaces für neue und vorhandene Tabellen zu aktivieren. Sie können die Konsole auch verwenden, um die Einstellungen für die automatische Skalierung zu ändern oder die automatische Skalierung zu deaktivieren.

### Note

Für erweiterte Funktionen wie das Festlegen von Ruhezeiten für das Abskalieren und Aufskalieren verwenden Sie CQL oder die AWS Command Line Interface (AWS CLI), um die Skalierungsrichtlinien für Amazon Keyspaces programmgesteuert zu verwalten. Weitere Informationen finden Sie unter [Verwalten von Amazon Keyspaces Auto Scaling mit Cassandra Query Language \(CQL\)](#) oder [Verwalten von Amazon Keyspaces-Skalierungsrichtlinien mit der CLI](#).

### Themen

- [Bevor Sie beginnen: Erteilen von Benutzerberechtigungen für Amazon Keyspaces Auto Scaling](#)
- [Erstellen einer neuen Tabelle mit aktivierter Amazon Keyspaces Auto Scaling](#)

- [Aktivieren der automatischen Skalierung von Amazon Keyspaces für vorhandene Tabellen](#)
- [Ändern oder Deaktivieren der automatischen Skalierungseinstellungen für Amazon Keyspaces](#)
- [Anzeigen von Auto-Scaling-Aktivitäten von Amazon Keyspaces in der Konsole](#)

## Bevor Sie beginnen: Erteilen von Benutzerberechtigungen für Amazon Keyspaces Auto Scaling

Vergewissern Sie sich zunächst, dass der Benutzer über die entsprechenden Berechtigungen zum Erstellen und Verwalten von Auto-Scaling-Einstellungen verfügt. In AWS Identity and Access Management (IAM) `AmazonKeyspacesFullAccess` ist die - AWS verwaltete Richtlinie erforderlich, um Amazon Keyspaces-Skalierungsrichtlinien zu verwalten.

### Important

`application-autoscaling:*`-Berechtigungen sind erforderlich, um die automatische Skalierung für eine Tabelle zu deaktivieren. Sie müssen die automatische Skalierung für eine Tabelle deaktivieren, bevor Sie sie löschen können.

Um einen IAM-Benutzer für den Zugriff auf die Amazon-Keyspaces-Konsole und die automatische Skalierung von Amazon Keyspaces einzurichten, fügen Sie die folgende Richtlinie hinzu.

So fügen Sie die **AmazonKeyspacesFullAccess** Richtlinie an

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 auf dem IAM-Konsolen-Dashboard Users (Benutzer) und anschließend Ihren IAM-Benutzer aus der Liste aus.
3. Wählen Sie auf der Seite Summary (Übersicht) die Option Add permissions (Berechtigungen hinzufügen) aus.
4. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen).
5. Wählen Sie in der Liste der Richtlinien `AmazonKeyspacesFullAccess` und dann Weiter: Überprüfen aus.
6. Wählen Sie Add permissions (Berechtigungen hinzufügen) aus.

## Erstellen einer neuen Tabelle mit aktivierter Amazon Keyspaces Auto Scaling

### Note

Die automatische Skalierung von Amazon Keyspaces erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), die automatische Skalierungsaktionen in Ihrem Namen durchführt. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

So erstellen Sie eine neue Tabelle mit aktivierter Auto Scaling

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tables (Tabellen) und anschließend Create table (Tabelle erstellen) aus.
3. Wählen Sie auf der Seite Tabelle erstellen im Abschnitt Tabellendetails einen Keyspace aus und geben Sie einen Namen für die neue Tabelle ein.
4. Erstellen Sie im Abschnitt Spalten das Schema für Ihre Tabelle.
5. Definieren Sie im Abschnitt Primärschlüssel den Primärschlüssel der Tabelle und wählen Sie optionale Clustering-Spalten aus.
6. Wählen Sie im Abschnitt Tabelleneinstellungen die Option Einstellungen anpassen aus.
7. Fahren Sie mit den Lese-/Schreibkapazitätseinstellungen fort.
8. Wählen Sie für Kapazitätsmodus die Option Bereitgestellt aus.
9. Vergewissern Sie sich im Abschnitt Lesekapazität, dass Automatische Skalierung ausgewählt ist.

In diesem Schritt wählen Sie die minimalen und maximalen Lesekapazitätseinheiten für die Tabelle sowie die Zielauslastung aus.

- Minimale Kapazitätseinheiten – Geben Sie den Wert für den Mindestdurchsatz ein, den die Tabelle immer unterstützen soll. Der Wert muss zwischen 1 und dem maximalen Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).

- Maximale Kapazitätseinheiten – Geben Sie die maximale Durchsatzmenge ein, die Sie für die Tabelle bereitstellen möchten. Der Wert muss zwischen 1 und dem maximalen Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
- Zielauslastung – Geben Sie eine Zielauslastungsrate zwischen 20 % und 90 % ein. Wenn der Datenverkehr die definierte Zielauslastungsrate überschreitet, wird die Kapazität automatisch hochskaliert. Wenn der Datenverkehr unter das definierte Ziel fällt, wird er automatisch wieder herunterskaliert.

#### Note

Weitere Informationen zu Standardkontingenten für Ihr Konto und deren Erhöhung finden Sie unter [Kontingente](#).

10. Wählen Sie im Abschnitt Schreibkapazität die gleichen Einstellungen aus, die im vorherigen Schritt für Lesekapazität definiert wurden, oder konfigurieren Sie Kapazitätswerte manuell.
11. Wählen Sie Create table (Tabelle erstellen) aus. Ihre Tabelle wird mit den angegebenen Auto-Scaling-Parametern erstellt.

## Aktivieren der automatischen Skalierung von Amazon Keyspaces für vorhandene Tabellen

#### Note

Die automatische Skalierung von Amazon Keyspaces erfordert das Vorhandensein einer serviceverknüpften Rolle (AWSServiceRoleForApplicationAutoScaling\_CassandraTable), die automatische Skalierungsaktionen in Ihrem Namen durchführt. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

So aktivieren Sie die automatische Skalierung von Amazon Keyspaces für eine vorhandene Tabelle

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.

2. Wählen Sie die Tabelle aus, mit der Sie arbeiten möchten, und wechseln Sie zur Registerkarte Kapazität.
3. Wählen Sie im Abschnitt Kapazitätseinstellungen die Option Bearbeiten aus.
4. Stellen Sie unter Kapazitätsmodus sicher, dass die Tabelle den Modus Bereitgestellte Kapazität verwendet.
5. Wählen Sie Automatisch skalieren und sehen Sie sich Schritt 6 in an [Erstellen einer neuen Tabelle mit aktivierter Amazon Keyspaces Auto Scaling](#), um die Lese- und Schreibkapazität zu bearbeiten.
6. Wenn die Einstellungen für die automatische Skalierung definiert sind, wählen Sie Speichern aus.

## Ändern oder Deaktivieren der automatischen Skalierungseinstellungen für Amazon Keyspaces

Sie können die verwenden AWS Management Console , um Ihre Einstellungen für die automatische Skalierung von Amazon Keyspaces zu ändern. Wählen Sie dazu die Tabelle aus, die Sie bearbeiten möchten, und wechseln Sie zur Registerkarte Kapazität. Wählen Sie im Abschnitt Kapazitätseinstellungen die Option Bearbeiten aus. Sie können jetzt die Einstellungen in den Abschnitten Lesekapazität oder Schreibkapazität ändern. Weitere Informationen zu diesen Einstellungen finden Sie unter [Erstellen einer neuen Tabelle mit aktivierter Amazon Keyspaces Auto Scaling](#).

Um die automatische Skalierung von Amazon Keyspaces zu deaktivieren, deaktivieren Sie das Kontrollkästchen Automatische Skalierung. Wenn Sie die automatische Skalierung deaktivieren, wird die Tabelle mit Application Auto Scaling als skalierbares Ziel abgemeldet. Um die serviceverknüpfte Rolle zu löschen, die von Application Auto Scaling für den Zugriff auf Ihre Amazon Keyspaces-Tabelle verwendet wird, führen Sie die Schritte unter aus [the section called “Löschen einer serviceverknüpften Rolle für Amazon Keyspaces”](#).

### Note

Um die serviceverknüpfte Rolle zu löschen, die Application Auto Scaling verwendet, müssen Sie die automatische Skalierung für alle Tabellen im Konto in allen deaktivieren AWS-Regionen.

## Anzeigen von Auto-Scaling-Aktivitäten von Amazon Keyspaces in der Konsole

Sie können überwachen, wie Amazon Keyspaces Auto Scaling Ressourcen nutzt, indem Sie Amazon CloudWatch verwenden, das Metriken über Ihre Nutzung und Leistung generiert. Führen Sie die Schritte im [Application Auto Scaling -Benutzerhandbuch](#) aus, um ein CloudWatch Dashboard zu erstellen.

## Verwalten von Amazon Keyspaces Auto Scaling mit Cassandra Query Language (CQL)

Um Auto-Scaling-Einstellungen für Amazon-Keyspaces-Tabellen mit Cassandra Query Language (CQL) zu erstellen und zu verwalten, können Sie verwenden `cqlsh`. Dieses Thema bietet einen Überblick über die Auto-Scaling-Aufgaben, die Sie programmgesteuert mit CQL verwalten können.

Weitere Informationen zu den in diesem Thema beschriebenen CQL-Anweisungen finden Sie unter [the section called “DDL-Anweisungen”](#).

### Themen

- [Bevor Sie beginnen](#)
- [Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe von CQL](#)
- [Aktivieren der automatischen Skalierung für eine vorhandene Tabelle mithilfe von CQL](#)
- [Anzeigen der Auto-Scaling-Konfiguration von Amazon Keyspaces Ihrer Tabelle mit CQL](#)
- [Deaktivieren von Amazon Keyspaces Auto Scaling für eine Tabelle mithilfe von CQL](#)

### Bevor Sie beginnen

Sie müssen die folgenden Aufgaben ausführen, bevor Sie beginnen können.

#### Konfigurieren von Berechtigungen

Wenn Sie dies noch nicht getan haben, müssen Sie die entsprechenden Berechtigungen für den Benutzer konfigurieren, um Auto-Scaling-Einstellungen zu erstellen und zu verwalten. In AWS Identity and Access Management (IAM) `AmazonKeyspacesFullAccess` ist die - AWS verwaltete Richtlinie erforderlich, um Amazon Keyspaces-Skalierungsrichtlinien zu verwalten. Die detaillierten Schritte finden Sie unter [the section called “Bevor Sie beginnen: Erteilen von Benutzerberechtigungen für Amazon Keyspaces Auto Scaling”](#).

## Konfigurieren von `cqlsh`

Falls noch nicht geschehen, müssen Sie installieren und konfigurieren `cqlsh`. Befolgen Sie dazu die Anweisungen unter [the section called “Verwenden des `cqlsh`-expansion”](#). Anschließend können Sie die verwenden AWS CloudShell , um die Befehle in den folgenden Abschnitten auszuführen.

## Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe von CQL

Wenn Sie eine neue Amazon Keyspaces-Tabelle erstellen, können Sie die automatische Skalierung für die Schreib- oder Lesekapazität der Tabelle in der `-CREATE TABLE`Anweisung automatisch aktivieren. Auf diese Weise kann Amazon Keyspaces Application Auto Scaling in Ihrem Namen kontaktieren, um die Tabelle als skalierbares Ziel zu registrieren und die bereitgestellte Schreib- oder Lesekapazität anzupassen.

Weitere Informationen zum Erstellen einer multiregionalen Tabelle und zum Konfigurieren verschiedener Auto-Scaling-Einstellungen für Tabellenreplikate finden Sie unter [the section called “Erstellen einer multiregionalen Tabelle mit Standardeinstellungen \(CQL\)”](#).

### Note

Amazon Keyspaces Auto Scaling erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), um automatische Skalierungsaktionen in Ihrem Namen durchzuführen. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

Um Auto-Scaling-Einstellungen für eine Tabelle programmgesteuert zu konfigurieren, verwenden Sie die `-AUTOSCALING_SETTINGS`Anweisung, die die Parameter für Amazon Keyspaces Auto Scaling enthält. Die Parameter definieren die Bedingungen, die Amazon Keyspaces anweisen, den bereitgestellten Durchsatz Ihrer Tabelle anzupassen, und welche zusätzlichen optionalen Aktionen ausgeführt werden sollen. In diesem Beispiel definieren Sie die Auto-Scaling-Einstellungen für `mytable` .

Die Richtlinie enthält die folgenden Elemente:

- `AUTOSCALING_SETTINGS` – Gibt an, ob Amazon Keyspaces die Durchsatzkapazität in Ihrem Namen anpassen darf. Die folgenden Werte sind erforderlich:
  - `provisioned_write_capacity_autoscaling_update`:

- `minimum_units`
- `maximum_units`
- `provisioned_read_capacity_autoscaling_update`:
  - `minimum_units`
  - `maximum_units`
- `scaling_policy` – Amazon Keyspaces unterstützt die Zielverfolgungsrichtlinie. Um die Richtlinie für die Ziel-Nachverfolgung zu definieren, konfigurieren Sie die folgenden Parameter.
  - `target_value` – Amazon Keyspaces Auto Scaling stellt sicher, dass das Verhältnis von verbrauchter Kapazität zu bereitgestellter Kapazität bei oder nahe diesem Wert bleibt. Sie definieren `target_value` als Prozentsatz.
  - `disableScaleIn`: (Optional) Ein `boolean`, der angibt, ob für die Tabelle deaktiviert oder aktiviert `scale-in` ist. Dieser Parameter ist standardmäßig deaktiviert. Um zu aktivieren `scale-in`, setzen Sie den `boolean` Wert auf `FALSE`. Das bedeutet, dass die Kapazität für eine Tabelle in Ihrem Namen automatisch herunterskaliert wird.
  - `scale_out_cooldown` – Eine Scale-Out-Aktivität erhöht den bereitgestellten Durchsatz Ihrer Tabelle. Um eine Ruhephase für Scale-Out-Aktivitäten hinzuzufügen, geben Sie einen Wert in Sekunden für `anscale_out_cooldown`. Wenn Sie keinen Wert angeben, ist der Standardwert 0. Weitere Informationen zur Zielverfolgung und Ruhephasen finden Sie unter [Skalierungsrichtlinien für die Zielverfolgung](#) im Benutzerhandbuch für Application Auto Scaling.
  - `scale_in_cooldown` – Eine Scale-In-Aktivität verringert den bereitgestellten Durchsatz Ihrer Tabelle. Um eine Ruhephase für Abskalierungsaktivitäten hinzuzufügen, geben Sie einen Wert in Sekunden für `anscale_in_cooldown`. Wenn Sie keinen Wert angeben, ist der Standardwert 0. Weitere Informationen zur Zielverfolgung und Ruhephasen finden Sie unter [Skalierungsrichtlinien für die Zielverfolgung](#) im Benutzerhandbuch für Application Auto Scaling.

#### Note

Um besser zu verstehen, wie der `target_value` funktioniert, nehmen Sie an, dass Sie über eine Tabelle mit einer Einstellung des bereitgestellten Durchsatzes von 200 Schreibkapazitätseinheiten verfügen. Sie entscheiden sich dafür, eine Skalierungsrichtlinie für diese Tabelle mit einem `target_value` von 70 % zu erstellen.

Angenommen, Sie beginnen den Schreibverkehr zu der Tabelle zu leiten, damit der tatsächliche Schreibdurchsatz bei 150 Kapazitätseinheiten liegt. Das `consumed-to-provisioned` Verhältnis ist jetzt (150/200) oder 75 Prozent. Dieses Verhältnis überschreitet



Ihr Ziel, sodass Auto Scaling die bereitgestellte Schreibkapazität auf 215 erhöht, sodass das Verhältnis (150/215) oder 69,77 Prozent ist – so nahe `target_value` wie möglich an Ihrem , aber nicht darüber.

Für `mytable` legen Sie sowohl für die Lese- als auch `TargetValue` für die Schreibkapazität auf 50 Prozent fest. Amazon Keyspaces Auto Scaling passt den bereitgestellten Durchsatz der Tabelle im Bereich von 5 bis 10 Kapazitätseinheiten an, sodass das `consumed-to-provisioned` Verhältnis bei oder nahe 50 Prozent bleibt. Für die Lesekapazität legen Sie die Werte für `ScaleOutCooldown` und `ScaleInCooldown` auf 60 Sekunden fest.

Sie können die folgende Anweisung verwenden, um eine neue Amazon Keyspaces-Tabelle mit aktiviertem Auto Scaling zu erstellen.

```
CREATE TABLE mykeyspace.mytable(pk int, ck int, PRIMARY KEY (pk, ck))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 }
}
```

```
};
```

## Aktivieren der automatischen Skalierung für eine vorhandene Tabelle mithilfe von CQL

Für eine vorhandene Amazon Keyspaces-Tabelle können Sie mithilfe der `-ALTER TABLE`-Anweisung die automatische Skalierung für die Schreib- oder Lesekapazität der Tabelle aktivieren. Wenn Sie eine Tabelle aktualisieren, die sich derzeit im On-Demand-Kapazitätsmodus befindet, `capacity_mode` ist dies erforderlich. Wenn sich Ihre Tabelle bereits im Modus bereitgestellter Kapazität befindet, kann dieses Feld weggelassen werden.

### Note

Die automatische Skalierung von Amazon Keyspaces erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), die automatische Skalierungsaktionen in Ihrem Namen durchführt. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

Im folgenden Beispiel aktualisiert die Anweisung die Tabelle `mytable`, die sich im On-Demand-Kapazitätsmodus befindet. Die Anweisung ändert den Kapazitätsmodus der Tabelle in den Modus bereitgestellter Kapazität mit aktivierter Auto Scaling.

Die Schreibkapazität wird im Bereich von 5 bis 10 Kapazitätseinheiten mit einem Zielwert von 50 % konfiguriert. Die Lesekapazität wird auch im Bereich von 5 bis 10 Kapazitätseinheiten mit einem Zielwert von 50 % konfiguriert. Für die Lesekapazität legen Sie die Werte für `scale_out_cooldown` und `scale_in_cooldown` auf 60 Sekunden fest.

```
ALTER TABLE mykeyspace.mytable
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
```

```

 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 }
 };

```

## Anzeigen der Auto-Scaling-Konfiguration von Amazon Keyspaces Ihrer Tabelle mit CQL

Verwenden Sie den folgenden Befehl, um Details zur Auto-Scaling-Konfiguration einer Tabelle anzuzeigen.

```

SELECT * FROM system_schema_mcs.autoscaling WHERE keyspace_name = 'mykeyspace' AND
table_name = 'mytable';

```

Die Ausgabe für diesen Befehl sieht wie folgt aus.

```

keyspace_name | table_name | provisioned_read_capacity_autoscaling_update
|
provisioned_write_capacity_autoscaling_update
-----+-----
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
mykeyspace | mytable | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 60, 'disable_scale_in': false, 'target_value':
50, 'scale_in_cooldown': 60}}} | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':

```

```
{'scale_out_cooldown': 0, 'disable_scale_in': false, 'target_value': 50,
'scale_in_cooldown': 0}}
```

## Deaktivieren von Amazon Keyspaces Auto Scaling für eine Tabelle mithilfe von CQL

Sie können die automatische Skalierung von Amazon Keyspaces für Ihre Tabelle jederzeit deaktivieren. Wenn Sie die Lese- oder Schreibkapazität Ihrer Tabelle nicht mehr skalieren müssen, sollten Sie die automatische Skalierung deaktivieren, damit Amazon Keyspaces die Lese- oder Schreibkapazitätseinstellungen Ihrer Tabelle nicht weiter ändert. Sie können die Tabelle mit einer `-ALTER TABLE`Anweisung aktualisieren.

Die folgende Anweisung deaktiviert die automatische Skalierung für die Schreibkapazität der Tabelle `mytable` . Außerdem werden die CloudWatch Alarme gelöscht, die in Ihrem Namen erstellt wurden.

```
ALTER TABLE mykeyspace.mytable
WITH AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'autoscaling_disabled': true
 }
};
```

### Note

Um die serviceverknüpfte Rolle zu löschen, die Application Auto Scaling verwendet, müssen Sie die automatische Skalierung für alle Tabellen im Konto in allen deaktivieren AWS-Regionen.

## Verwalten von Amazon Keyspaces-Skalierungsrichtlinien mit der CLI

Um die Auto-Scaling-Einstellungen von Amazon Keyspaces programmgesteuert zu aktualisieren und zu verwalten, können Sie die AWS Command Line Interface (AWS CLI) oder die AWS -API verwenden. Informationen zum Verwalten von Auto-Scaling-Richtlinien für Amazon Keyspaces mit Cassandra Query Language (CQL) finden Sie unter [the section called “Verwenden von CQL”](#) . Dieses Thema bietet einen Überblick über die Auto-Scaling-Aufgaben, die Sie programmgesteuert mithilfe der verwalten können AWS CLI.

Weitere Informationen zu den in diesem Thema beschriebenen Amazon Keyspaces- AWS CLI Befehlen finden Sie in der [-AWS CLI Befehlsreferenz](#).

## Themen

- [Bevor Sie beginnen](#)
- [Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe der AWS CLI](#)
- [Aktivieren der automatischen Skalierung für eine vorhandene Tabelle mithilfe der AWS CLI](#)
- [Anzeigen der Auto-Scaling-Konfiguration von Amazon Keyspaces Ihrer Tabelle mithilfe der AWS CLI](#)
- [Deaktivieren der automatischen Skalierung von Amazon Keyspaces für eine Tabelle mithilfe der AWS CLI](#)

## Bevor Sie beginnen

Sie müssen die folgenden Aufgaben ausführen, bevor Sie beginnen können.

### Konfigurieren von Berechtigungen

Wenn Sie dies noch nicht getan haben, müssen Sie die entsprechenden Berechtigungen für den Benutzer konfigurieren, um Auto-Scaling-Einstellungen zu erstellen und zu verwalten. In AWS Identity and Access Management (IAM) `AmazonKeyspacesFullAccess` ist die - AWS verwaltete Richtlinie erforderlich, um Amazon Keyspaces-Skalierungsrichtlinien zu verwalten. Die detaillierten Schritte finden Sie unter [the section called “Bevor Sie beginnen: Erteilen von Benutzerberechtigungen für Amazon Keyspaces Auto Scaling”](#).

### AWS CLI installieren

Wenn Sie es noch nicht getan haben, müssen Sie AWS CLI installieren und konfigurieren. Gehen Sie dazu zum [AWS Command Line Interface -Benutzerhandbuch](#) und folgen Sie diesen Anweisungen:

- [Installieren des AWS CLI](#)
- [Konfigurieren von AWS CLI](#)

## Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe der AWS CLI

Wenn Sie eine neue Amazon Keyspaces-Tabelle erstellen, können Sie die automatische Skalierung für die Schreib- oder Lesekapazität der Tabelle in der `-CreateTableOperation` automatisch aktivieren. Auf diese Weise kann Amazon Keyspaces Application Auto Scaling in Ihrem Namen kontaktieren, um die Tabelle zu registrieren, die Sie als skalierbares Ziel angeben, und die bereitgestellte Schreib- oder Lesekapazität anpassen.

Weitere Informationen zum Erstellen einer multiregionalen Tabelle mit Auto-Scaling-Konfiguration finden Sie unter [the section called “Erstellen einer neuen multiregionalen Tabelle im Bereitstellungsmodus mit Auto Scaling \(CLI\)”](#).

### Note

Amazon Keyspaces Auto Scaling erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), um automatische Skalierungsaktionen in Ihrem Namen durchzuführen. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

Um Auto-Scaling-Einstellungen für eine Tabelle programmgesteuert zu konfigurieren, verwenden Sie die `-autoScalingSpecification`Aktion, die die Parameter für Amazon Keyspaces Auto Scaling definiert. Die Parameter definieren die Bedingungen, die Amazon Keyspaces anweisen, den bereitgestellten Durchsatz Ihrer Tabelle anzupassen, und welche zusätzlichen optionalen Aktionen ausgeführt werden sollen. In diesem Beispiel definieren Sie die Auto-Scaling-Einstellungen für `mytable`.

Die Richtlinie enthält die folgenden Elemente:

- `autoScalingSpecification` – Gibt an, ob Amazon Keyspaces den Kapazitätsdurchsatz in Ihrem Namen anpassen darf. Sie können Auto Scaling für Lese- und Schreibkapazität separat aktivieren. Anschließend müssen Sie die folgenden Parameter für `autoScalingSpecification` angeben:
  - `writeCapacityAutoScaling` – Die maximale und minimale Schreibkapazitätseinheiten.
  - `readCapacityAutoScaling` – Die maximale und minimale Lesekapazitätseinheiten.
  - `scalingPolicy` – Amazon Keyspaces unterstützt die Zielverfolgungsrichtlinie. Um die Richtlinie für die Ziel-Nachverfolgung zu definieren, konfigurieren Sie die folgenden Parameter.
    - `targetValue` – Amazon Keyspaces Auto Scaling stellt sicher, dass das Verhältnis von verbrauchter Kapazität zu bereitgestellter Kapazität bei oder nahe diesem Wert bleibt. Sie definieren `targetValue` als Prozentsatz.
    - `disableScaleIn`: (Optional) Ein boolean, der angibt, ob für die Tabelle deaktiviert oder aktiviert `scale-in` ist. Dieser Parameter ist standardmäßig deaktiviert. Um zu aktivieren `scale-in`, setzen Sie den boolean Wert auf `FALSE`. Das bedeutet, dass die Kapazität für eine Tabelle in Ihrem Namen automatisch herunterskaliert wird.

- `scaleOutCooldown` – Eine Scale-Out-Aktivität erhöht den bereitgestellten Durchsatz Ihrer Tabelle. Um eine Ruhephase für Scale-Out-Aktivitäten hinzuzufügen, geben Sie einen Wert in Sekunden für `anScaleOutCooldown`. Der Standardwert lautet 0. Weitere Informationen zur Ziel-Nachverfolgung und Ruhephasen finden Sie unter [Skalierungsrichtlinien für die Ziel-Nachverfolgung](#) im Benutzerhandbuch für Application Auto Scaling.
- `scaleInCooldown` – Eine Scale-In-Aktivität verringert den bereitgestellten Durchsatz Ihrer Tabelle. Um eine Ruhephase für Abskalierungsaktivitäten hinzuzufügen, geben Sie einen Wert in Sekunden für `anScaleInCooldown`. Der Standardwert lautet 0. Weitere Informationen zur Zielverfolgung und Ruhephasen finden Sie unter [Skalierungsrichtlinien für die Zielverfolgung](#) im Benutzerhandbuch für Application Auto Scaling.

### Note

Um besser zu verstehen, wie der `TargetValue` funktioniert, nehmen Sie an, dass Sie über eine Tabelle mit einer Einstellung des bereitgestellten Durchsatzes von 200 Schreibkapazitätseinheiten verfügen. Sie entscheiden sich dafür, eine Skalierungsrichtlinie für diese Tabelle mit einem `TargetValue` von 70 % zu erstellen.

Angenommen, Sie beginnen den Schreibverkehr zu der Tabelle zu leiten, damit der tatsächliche Schreibdurchsatz bei 150 Kapazitätseinheiten liegt. Das `consumed-to-provisioned` Verhältnis ist jetzt (150/200) oder 75 Prozent. Dieses Verhältnis überschreitet Ihr Ziel, sodass Auto Scaling die bereitgestellte Schreibkapazität auf 215 erhöht, sodass das Verhältnis (150/215) oder 69,77 Prozent ist – so nahe `TargetValue` wie möglich an Ihrem , aber nicht darüber.

Für `mytable` legen Sie `TargetValue` für Lese- und Schreibkapazität auf 50 Prozent fest. Amazon Keyspaces Auto Scaling passt den bereitgestellten Durchsatz der Tabelle im Bereich von 5 bis 10 Kapazitätseinheiten an, sodass das `consumed-to-provisioned` Verhältnis bei oder nahe 50 Prozent bleibt. Für die Lesekapazität legen Sie die Werte für `ScaleOutCooldown` und `ScaleInCooldown` auf 60 Sekunden fest.

Beim Erstellen von Tabellen mit komplexen Auto-Scaling-Einstellungen ist es hilfreich, die Auto-Scaling-Einstellungen aus einer JSON-Datei zu laden. Im folgenden Beispiel können Sie die Beispiel-JSON-Datei von [auto-scaling.zip](#) herunterladen und extrahieren `auto-scaling.json`, wobei Sie den Pfad zur Datei notieren. In diesem Beispiel befindet sich die JSON-Datei im aktuellen

Verzeichnis. Informationen zu verschiedenen Dateipfadoptionen finden Sie unter [So laden Sie Parameter aus einer Datei](#).

```
aws keyspaces create-table --keyspace-name mykeyspace --table-name mytable
 \ --schema-definition 'allColumns=[{name=pk,type=int},
{name=ck,type=int}],partitionKeys=[{name=pk},{name=ck}]'
 \ --capacity-specification
throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --auto-scaling-specification file://auto-scaling.json
```

## Aktivieren der automatischen Skalierung für eine vorhandene Tabelle mithilfe der AWS CLI

Für eine vorhandene Amazon Keyspaces-Tabelle können Sie mithilfe der `-updateTableOperation` die automatische Skalierung für die Schreib- oder Lesekapazität der Tabelle aktivieren. Weitere Informationen zum Aktualisieren der Auto-Scaling-Einstellungen für eine Tabelle mit mehreren Regionen finden Sie unter [the section called “Aktualisieren der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen \(CLI\)”](#).

### Note

Die automatische Skalierung von Amazon Keyspaces erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), die automatische Skalierungsaktionen in Ihrem Namen durchführt. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

Sie können den folgenden Befehl verwenden, um die automatische Skalierung von Amazon Keyspaces für eine vorhandene Tabelle zu aktivieren. Die Auto-Scaling-Einstellungen für die Tabelle werden aus einer JSON-Datei geladen. Im folgenden Beispiel können Sie die Beispiel-JSON-Datei von [auto-scaling.zip](#) herunterladen und extrahieren `auto-scaling.json`, wobei Sie den Pfad zur Datei notieren. In diesem Beispiel befindet sich die JSON-Datei im aktuellen Verzeichnis. Informationen zu verschiedenen Dateipfadoptionen finden Sie unter [So laden Sie Parameter aus einer Datei](#).



Weitere Informationen zu den im folgenden Beispiel verwendeten Auto-Scaling-Einstellungen finden Sie unter [the section called “Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe der AWS CLI”](#).

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --capacity-specification
throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --auto-scaling-specification file://auto-scaling.json
```

## Anzeigen der Auto-Scaling-Konfiguration von Amazon Keyspaces Ihrer Tabelle mithilfe der AWS CLI

Um die Auto-Scaling-Konfiguration einer Tabelle anzuzeigen, können Sie die `-get-table-auto-scaling-settings` Operation verwenden. Der folgende CLI-Befehl ist ein Beispiel dafür.

```
aws keyspaces get-table-auto-scaling-settings --keyspace-name mykeyspace --table-name
mytable
```

Die Ausgabe für diesen Befehl sieht wie folgt aus.

```
{
 "keyspaceName": "mykeyspace",
 "tableName": "mytable",
 "resourceArn": "arn:aws:cassandra:us-east-1:5555-5555-5555:/keyspace/mykeyspace/
table/mytable",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
```

```
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 50.0
 }
 }
 }
}
```

## Deaktivieren der automatischen Skalierung von Amazon Keyspaces für eine Tabelle mithilfe der AWS CLI

Sie können die automatische Skalierung von Amazon Keyspaces für Ihre Tabelle jederzeit deaktivieren. Wenn Sie die Lese- oder Schreibkapazität Ihrer Tabelle nicht mehr skalieren müssen, sollten Sie die automatische Skalierung deaktivieren, damit Amazon Keyspaces die Lese- oder Schreibkapazitätseinstellungen Ihrer Tabelle nicht weiter ändert. Sie können die Tabelle mit einer `-UpdateTableOperation` aktualisieren.

Mit dem folgenden Befehl wird die automatische Skalierung für die Lesekapazität der Tabelle deaktiviert. Außerdem werden die CloudWatch Alarmlisten gelöscht, die in Ihrem Namen erstellt wurden.

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --auto-scaling-specification
 readCapacityAutoScaling={autoScalingDisabled=true}
```

### Note

Um die serviceverknüpfte Rolle zu löschen, die Application Auto Scaling verwendet, müssen Sie die automatische Skalierung für alle Tabellen im Konto in allen deaktivieren AWS-Regionen.

## Effektive Nutzung von Burst-Kapazität in Amazon Keyspaces

Amazon Keyspaces bietet eine gewisse Flexibilität bei der Durchsatzbereitstellung pro Partition, indem es Burst-Kapazität bereitstellt. Immer wenn Sie den Durchsatz einer Partition nicht vollständig

nutzen, reserviert Amazon Keyspaces einen Teil dieser ungenutzten Kapazität für spätere Durchsatzspitzen, um Nutzungsspitzen zu bewältigen.

Amazon Keyspaces behält derzeit bis zu 5 Minuten (300 Sekunden) ungenutzte Lese- und Schreibkapazität. Bei gelegentlichen Lese- oder Schreibaktivitäten können diese zusätzlichen Kapazitätseinheiten schnell verbraucht werden — sogar schneller als die pro Sekunde bereitgestellte Durchsatzkapazität, die Sie für Ihre Tabelle definiert haben.

Amazon Keyspaces kann auch ohne vorherige Ankündigung Burst-Kapazität für die Hintergrundwartung und andere Aufgaben verbrauchen.

Beachten Sie, dass die Burst-Kapazität in Zukunft geändert werden kann.

## So schätzen Sie den Kapazitätsverbrauch in Amazon Keyspaces

Wenn Sie Daten in Amazon Keyspaces lesen oder schreiben, hängt die Menge der Lese-/Schreibanforderungseinheiten (RRUs/WRUs) oder Lese-/Schreibkapazitätseinheiten (RCUs/WCUs), die Ihre Abfrage verbraucht, von der Gesamtmenge der Daten ab, die Amazon Keyspaces verarbeiten muss, um die Abfrage auszuführen. In einigen Fällen können die an den Client zurückgegebenen Daten eine Teilmenge der Daten sein, die Amazon Keyspaces lesen musste, um die Abfrage zu verarbeiten. Bei bedingten Schreibvorgängen verbraucht Amazon Keyspaces Schreibkapazität, auch wenn die bedingte Prüfung fehlschlägt.

Um die Gesamtmenge der für eine Anfrage verarbeiteten Daten abzuschätzen, müssen Sie die codierte Größe einer Zeile und die Gesamtzahl der Zeilen berücksichtigen. In diesem Thema werden einige Beispiele für gängige Szenarien und Zugriffsmuster behandelt, um zu zeigen, wie Amazon Keyspaces Abfragen verarbeitet und wie sich dies auf den Kapazitätsverbrauch auswirkt. Sie können den Beispielen folgen, um den Kapazitätsbedarf Ihrer Tabellen abzuschätzen, und Amazon verwenden, CloudWatch um den Lese- und Schreibkapazitätsverbrauch für diese Anwendungsfälle zu beobachten.

Informationen zur Berechnung der codierten Zeilengröße in Amazon Keyspaces finden Sie unter [the section called “Berechnung der Zeilengröße”](#)

### Themen

- [Bereichsabfragen](#)
- [Abfragen einschränken](#)
- [Tabellenscans](#)

- [Leichte Transaktionen](#)
- [Schätzen Sie den Lese- und Schreibkapazitätsverbrauch mit Amazon CloudWatch](#)

## Bereichsabfragen

Um den Lesekapazitätsverbrauch einer Bereichsabfrage zu untersuchen, verwenden wir die folgende Beispieldatenbank, die den On-Demand-Kapazitätsmodus verwendet.

```
pk1 | pk2 | pk3 | ck1 | ck2 | ck3 | value
-----+-----+-----+-----+-----+-----+-----
a | b | 1 | a | b | 50 | <any value that results in a row size larger than 4KB>
a | b | 1 | a | b | 60 | value_1
a | b | 1 | a | b | 70 | <any value that results in a row size larger than 4KB>
```

Führen Sie nun die folgende Abfrage für diese Tabelle aus.

```
SELECT * FROM amazon_keyspaces.example_table_1 WHERE pk1='a' AND pk2='b' AND pk3=1 AND
ck1='a' AND ck2='b' AND ck3 > 50 AND ck3 < 70;
```

Sie erhalten die folgende Ergebnismenge aus der Abfrage und der von Amazon Keyspaces ausgeführte Lesevorgang verbraucht 2 RRUs im LOCAL\_QUORUM Konsistenzmodus.

```
pk1 | pk2 | pk3 | ck1 | ck2 | ck3 | value
-----+-----+-----+-----+-----+-----+-----
a | b | 1 | a | b | 60 | value_1
```

Amazon Keyspaces verwendet 2 RRUs, um die Zeilen mit den Werten auszuwerten  $ck3=60$  und die Abfrage  $ck3=70$  zu verarbeiten. Amazon Keyspaces gibt jedoch nur die Zeile zurück, in der die in der Abfrage angegebene WHERE Bedingung wahr ist, also die Zeile mit dem Wert  $ck3=60$ . Um den in der Abfrage angegebenen Bereich auszuwerten, liest Amazon Keyspaces in diesem Fall die Zeile, die der Obergrenze des Bereichs entspricht  $ck3 = 70$ , gibt diese Zeile jedoch nicht im Ergebnis zurück. Der Lesekapazitätsverbrauch basiert auf den bei der Verarbeitung der Abfrage gelesenen Daten, nicht auf den zurückgegebenen Daten.

## Abfragen einschränken

Bei der Verarbeitung einer Abfrage, die die LIMIT Klausel verwendet, liest Amazon Keyspaces Zeilen bis zur maximalen Seitengröße, wenn versucht wird, die in der Abfrage angegebene

Bedingung zu erfüllen. Wenn Amazon Keyspaces nicht genügend übereinstimmende Daten finden kann, die dem LIMIT Wert auf der ersten Seite entsprechen, sind möglicherweise ein oder mehrere paginierte Aufrufe erforderlich. Um mit dem Lesen auf der nächsten Seite fortzufahren, können Sie ein Paginierungstoken verwenden. Die Standard-Seitengröße ist 1 MB. Um bei der Verwendung von LIMIT Klauseln weniger Lesekapazität zu verbrauchen, können Sie die Seitengröße reduzieren. Weitere Hinweise zur Seitennummerierung finden Sie unter [the section called “Ergebnisse paginieren”](#)

Schauen wir uns als Beispiel die folgende Abfrage an.

```
SELECT * FROM my_table WHERE partition_key=1234 LIMIT 1;"
```

Wenn Sie die Seitengröße nicht festlegen, liest Amazon Keyspaces 1 MB an Daten, obwohl es Ihnen nur 1 Zeile zurückgibt. Damit Amazon Keyspaces nur eine Zeile liest, können Sie die Seitengröße für diese Abfrage auf 1 setzen. In diesem Fall würde Amazon Keyspaces nur eine Zeile lesen, vorausgesetzt, Sie haben keine abgelaufenen Zeilen, die auf ime-to-live T-Einstellungen oder clientseitigen Zeitstempeln basieren. Um weniger Lesekapazität zu verbrauchen, empfehlen wir, Ihre Seitengröße auf den LIMIT Wert einzustellen, um die von Amazon Keyspaces gelesene Datenmenge zu reduzieren.

## Tabellenscans

Abfragen, die zu vollständigen Tabellenscans führen, z. B. Abfragen, die die ALLOW FILTERING Option verwenden, sind ein weiteres Beispiel für Abfragen, die mehr Lesevorgänge verarbeiten, als sie als Ergebnisse zurückgeben. Und der Lesekapazitätsverbrauch basiert auf den gelesenen Daten, nicht auf den zurückgegebenen Daten.

Für das Beispiel für den Tabellenscan verwenden wir die folgende Beispieltabelle im On-Demand-Kapazitätsmodus.

```
pk | ck | value
----+-----+-----
pk | 10 | <any value that results in a row size larger than 4KB>
pk | 20 | value_1
pk | 30 | <any value that results in a row size larger than 4KB>
```

Amazon Keyspaces erstellt standardmäßig eine Tabelle im On-Demand-Kapazitätsmodus mit vier Partitionen. In dieser Beispieltabelle werden alle Daten in einer Partition gespeichert und die restlichen drei Partitionen sind leer.

Führen Sie nun die folgende Abfrage für die Tabelle aus.

```
SELECT * from amazon_keyspaces.example_table_2;
```

Diese Abfrage führt zu einem Tabellenscanvorgang, bei dem Amazon Keyspaces alle vier Partitionen der Tabelle scannt und im LOCAL\_QUORUM Konsistenzmodus 6 RRUUs verbraucht. Erstens verbraucht Amazon Keyspaces 3 RRUUs zum Lesen der drei Zeilen mit `pk='pk'`. Dann verbraucht Amazon Keyspaces die zusätzlichen 3 RRUUs zum Scannen der drei leeren Partitionen der Tabelle. Da diese Abfrage zu einem Tabellenscan führt, scannt Amazon Keyspaces alle Partitionen in der Tabelle, einschließlich Partitionen ohne Daten.

## Leichte Transaktionen

Lightweight Transactions (LWT) ermöglichen es Ihnen, bedingte Schreiboperationen für Ihre Tabellendaten durchzuführen. Bedingte Aktualisierungsoperationen sind nützlich, wenn Datensätze auf der Grundlage von Bedingungen eingefügt, aktualisiert und gelöscht werden, die den aktuellen Status bewerten.

In Amazon Keyspaces erfordern alle Schreibvorgänge die Konsistenz LOCAL\_QUORUM, und es fallen keine zusätzlichen Gebühren für die Verwendung von LWTs an. Der Unterschied bei LWTs besteht darin, dass, wenn eine LWT-Zustandsprüfung den Wert FALSE ergibt, Schreibkapazitätseinheiten verbraucht werden. Die Anzahl der verbrauchten Schreibkapazitätseinheiten hängt von der Größe der Zeile ab. Wenn die Zeilengröße 2 KB beträgt, verbraucht das fehlgeschlagene bedingte Schreiben zwei Schreibkapazitätseinheiten. Wenn die Zeile derzeit nicht in der Tabelle vorhanden ist, verbraucht der Vorgang eine Schreibkapazitätseinheit. Durch die Überwachung der angegebenen ConditionalCheckFailed Metrik können CloudWatch Sie die Kapazität ermitteln, die durch Fehlschläge bei der LWT-Zustandsprüfung verbraucht wird.

## Schätzen Sie den Lese- und Schreibkapazitätsverbrauch mit Amazon CloudWatch

Um den Lese- und Schreibkapazitätsverbrauch abzuschätzen und zu überwachen, können Sie ein CloudWatch Dashboard verwenden. Weitere Informationen zu verfügbaren Metriken für Amazon Keyspaces finden Sie unter [the section called “Metriken und Dimensionen”](#).

Gehen Sie wie folgt vor, um die Lese- und Schreibkapazitätseinheiten zu überwachen CloudWatch, die von einer bestimmten Anweisung mit verbraucht werden.

1. Erstellen Sie eine neue Tabelle mit Beispieldaten

2. Konfigurieren Sie ein Amazon CloudWatch Keyspaces-Dashboard für die Tabelle. Zu Beginn können Sie eine Dashboard-Vorlage verwenden, die auf [Github](#) verfügbar ist.
3. Führen Sie die CQL-Anweisung aus, beispielsweise mit der `ALLOW FILTERING` Option, und überprüfen Sie im Dashboard, welche Lesekapazitätseinheiten für den vollständigen Tabellenscan verbraucht wurden.

# Arbeiten mit Keyspaces, Tabellen und Zeilen in Amazon Keyspaces (für Apache Cassandra)

Dieses Kapitel enthält Details zur Arbeit mit Keyspaces, Tabellen, Zeilen und mehr in Amazon Keyspaces (für Apache Cassandra). Informationen zur Überwachung von Keyspaces und Tabellen mit Amazon CloudWatch finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

## Themen

- [Arbeiten mit Keyspaces in Amazon Keyspaces](#)
- [Arbeiten mit Tabellen in Amazon Keyspaces](#)
- [Arbeiten mit Zeilen in Amazon Keyspaces](#)
- [Arbeiten mit Abfragen in Amazon Keyspaces](#)
- [Arbeiten mit Partitionierern in Amazon Keyspaces](#)
- [Arbeiten mit Tags und Labels für Amazon Keyspaces-Ressourcen](#)

## Arbeiten mit Keyspaces in Amazon Keyspaces

Dieser Abschnitt enthält Details zum Arbeiten mit Keyspaces in Amazon Keyspaces (für Apache Cassandra).

## Themen

- [Arbeiten mit System-Keyspaces in Amazon Keyspaces](#)
- [Erstellen von Keyspaces in Amazon Keyspaces](#)

## Arbeiten mit System-Keyspaces in Amazon Keyspaces

Amazon Keyspaces verwendet vier System-Keyspaces:

- system
- system\_schema
- system\_schema\_mcs
- system\_multiregion\_info



Die folgenden Abschnitte enthalten Details zu den System-Keyspaces und den Systemtabellen, die in Amazon Keyspaces unterstützt werden.

## system

Dies ist ein Cassandra-Schlüsselraum. Amazon Keyspaces verwendet die folgenden Tabellen.

| Tabellennamen | Spaltennamen                                                                                                                                                                                                                                                                                                 | Kommentare                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| local         | key, bootstrap<br>ped, broadcast<br>_address, cluster_n<br>ame, cql_versi<br>on, data_cent<br>er, gossip_ge<br>neration, host_id,<br>listen_address,<br>native_protocol_ve<br>rsion, partition<br>er, rack, release_v<br>ersion, rpc_addre<br>ss, schema_version,<br>thrift_version,<br>tokens, truncated_at | Informationen über den<br>lokalen Keyspace.                                                                                                                                                                                                                                                                                                                                                         |
| peers         | peer, data_center,<br>host_id, preferred<br>_ip, rack, release_v<br>ersion, rpc_addre<br>ss, schema_version,<br>tokens                                                                                                                                                                                       | Fragen Sie diese Tabelle ab,<br>um die verfügbaren Endpunkte<br>anzuzeigen. Wenn Sie<br>beispielsweise eine Verbindun<br>g über einen öffentlichen<br>Endpunkt herstellen, wird<br>eine Liste mit neun verfügbar<br>en IP-Adressen angezeigt.<br>Wenn Sie eine Verbindung<br>über einen FIPS-Endpunkt<br>herstellen, wird eine Liste mit<br>drei IP-Adressen angezeigt<br>. Wenn Sie eine Verbindun |

| Tabellennamen       | Spaltennamen                                                                                      | Kommentare                                                                                                                                                                                                                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |                                                                                                   | g über einen AWS PrivateLink VPC-Endpunkt herstellen, wird die Liste der IP-Adressen angezeigt, die Sie konfiguriert haben. Weitere Informationen finden Sie unter <a href="#">the section called “Auffüllen von system.peers Tabelleneinträgen mit VPC-Endpunktinformationen der Schnittstelle”</a> . |
| size_estimates      | keyspace_name,<br>table_name, range_start, range_end,<br>mean_partition_size,<br>partitions_count | Diese Tabelle definiert die Gesamtgröße und die Anzahl der Partitionen für jeden Tokenbereich für jede Tabelle. Dies ist für den Apache Cassandra Spark Connector erforderlich, der die geschätzte Partitionsgröße verwendet, um die Arbeit zu verteilen.                                              |
| prepared_statements | prepared_id,<br>logged_keyspace,<br>query_string                                                  | Diese Tabelle enthält Informationen zu gespeicherten Abfragen.                                                                                                                                                                                                                                         |

## system\_schema

Dies ist ein Cassandra-Schlüsselraum. Amazon Keyspaces verwendet die folgenden Tabellen.

| Tabellennamen | Spaltennamen                                     | Kommentare                                  |
|---------------|--------------------------------------------------|---------------------------------------------|
| keyspaces     | keyspace_name,<br>durable_writes,<br>replication | Informationen zu einem bestimmten Keyspace. |

| Tabellennamen | Spaltennamen                                                                                                                                                                                                                                                                                                          | Kommentare                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| tables        | keyspace_name, table_name, bloom_filter_fp_chance, caching, comment, compaction, compression, crc_check_chance, dclocal_read_repair_chance, default_time_to_live, extensions, flags, gc_grace_seconds, id, max_index_interval, memtable_flush_period_in_ms, min_index_interval, read_repair_chance, speculative_retry | Informationen zu einer bestimmten Tabelle. |
| columns       | keyspace_name, table_name, column_name, clustering_order, column_name_bytes, kind, position, type                                                                                                                                                                                                                     | Informationen zu einer bestimmten Spalte.  |

## system\_schema\_mcs

Dies ist ein Amazon Keyspaces Keyspace, der Informationen zu AWS oder Amazon Keyspaces spezifischen Einstellungen speichert.

| Tabellennamen  | Spaltennamen                                                                                                                                                                                                                                                                                                                                                                              | Kommentare                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| keyspaces      | keyspace_name,<br>durable_writes,<br>replication                                                                                                                                                                                                                                                                                                                                          | Fragen Sie diese Tabelle ab, um programmgesteuert herauszufinden, ob ein Keyspace erstellt wurde. Weitere Informationen finden Sie unter <a href="#">the section called “Erstellen von Keyspaces”</a> .                                                                                                                                                                                                                                                                                                                                                                           |
| tables         | keyspace_name,<br>creation_time,<br>speculative_retry,<br>cdc, gc_grace_ seconds, crc_check_ chance, min_index_ interval, bloom_fil ter_fp_chance, flags, custom_pr operties, dclocal_r ead_repair_chance, table_name, caching, default_time_to_li ve, read_repa ir_chance, max_index_ interval, extension s, compaction, comment, id, compressi on, memtable_ flush_period_in_ms, status | <p>Fragen Sie diese Tabelle ab, um den Status einer bestimmten Tabelle zu ermitteln. Weitere Informati onen finden Sie unter <a href="#">the section called “Erstellen von Tabellen”</a>.</p> <p>Sie können diese Tabelle auch abfragen, um Einstellu ngen aufzulisten, die für Amazon Keyspaces spezifisc h sind und als gespeiche rt werdencustom_pr operties . Beispielsweise:</p> <ul style="list-style-type: none"> <li>• capacity_mode</li> <li>• client_side_timest amps</li> <li>• encryption_specifi cation</li> <li>• point_in_time_reco very</li> <li>• ttl</li> </ul> |
| tables_history | keyspace_name,<br>table_name, event_tim                                                                                                                                                                                                                                                                                                                                                   | Fragen Sie diese Tabelle ab, um mehr über Schemaänd                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Tabellennamen            | Spaltennamen                                                                                                                        | Kommentare                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | <code>e, creation_time, custom_properties, event</code>                                                                             | erungen für eine bestimmte Tabelle zu erfahren.                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>columns</code>     | <code>keyspace_name, table_name, column_name, clustering_order, column_name_bytes, kind, position, type</code>                      | Diese Tabelle ist mit der Cassandra-Tabelle im <code>system_schema</code> Keyspace identisch.                                                                                                                                                                                                                                                                                                                                                |
| <code>tags</code>        | <code>resource_id, keyspace_name, resource_name, resource_type, tags</code>                                                         | Fragen Sie diese Tabelle ab, um herauszufinden, ob ein Keyspace Tags hat. Weitere Informationen finden Sie unter <a href="#">the section called “Hinzufügen von von Spaces zu neuen oder vorhandenen Schlüsselräumen und Tabellen mit CQL”</a> .                                                                                                                                                                                             |
| <code>autoscaling</code> | <code>keyspace_name, table_name, provisioned_read_capacity_autoscaling_update, provisioned_write_capacity_autoscaling_update</code> | Fragen Sie diese Tabelle ab, um die Auto-Scaling-Einstellungen einer bereitgestellten Tabelle abzurufen. Beachten Sie, dass diese Einstellungen erst verfügbar sind, wenn die Tabelle aktiv ist. Um diese Tabelle abzufragen, müssen Sie <code>keyspace_name</code> und <code>table_name</code> in der <code>-WHERE</code> -Klausel angeben. Weitere Informationen finden Sie unter <a href="#">the section called “Verwenden von CQL”</a> . |

## system\_multiregion\_info

Dies ist ein Amazon Keyspaces Keyspace, der Informationen über die Multi-Region-Replikation speichert.

| Tabellennamen | Spaltennamen                                                                                        | Kommentare                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tables        | keyspace_name,<br>table_name, region,<br>status                                                     | <p>Diese Tabelle enthält Informationen über multiregionale Tabellen, z. B. die , in der AWS-Regionen die Tabelle repliziert wird, und den Status der Tabelle. Sie können diese Tabelle auch abfragen, um Einstellungen aufzulisten, die spezifisch für Amazon Keyspaces sind, die als gespeichert werden <code>custom_properties</code> . Beispielsweise:</p> <ul style="list-style-type: none"> <li>• <code>capacity_mode</code></li> </ul> <p>Um diese Tabelle abzufragen, müssen Sie <code>keyspace_name</code> und <code>table_name</code> in der <code>-WHERE</code>-Klausel angeben. Weitere Informationen finden Sie unter <a href="#">the section called “Erstellen eines multiregionalen Keyspace (CQL)”</a>.</p> |
| autoscaling   | keyspace_name,<br>table_name, provisioned_read_capacity_automatically_update,<br>provisioned_write_ | <p>Fragen Sie diese Tabelle ab, um die Auto-Scaling-Einstellungen einer multiregionalen bereitgestellten Tabelle abzurufen. Beachten Sie,</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Tabellennamen | Spaltennamen                        | Kommentare                                                                                                                                                                                                                                                                    |
|---------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | capacity_autoscaling_update, region | dass diese Einstellungen erst verfügbar sind, wenn die Tabelle aktiv ist. Um diese Tabelle abzufragen, müssen Sie keyspace_name und table_name in der -WHERE-Klausel angeben. Weitere Informationen finden Sie unter <a href="#">the section called "Verwenden von CQL"</a> . |

## Erstellen von Keyspaces in Amazon Keyspaces

Amazon Keyspaces führt asynchron Data Definition Language (DDL)-Operationen wie das Erstellen und Löschen von Keyspaces durch.

Sie können den Erstellungsstatus neuer Keyspaces in der überwachten AWS Management Console, was angibt, wann ein Keyspace aussteht oder aktiv ist. Sie können den Erstellungsstatus eines neuen Keyspace auch programmgesteuert überwachen, indem Sie den system\_schema\_mcs Keyspace verwenden. Ein Keyspace wird in der system\_schema\_mcs keyspaces Tabelle sichtbar, wenn er einsatzbereit ist.

Das empfohlene Entwurfsmuster, das überprüft, wann ein neuer Keyspace einsatzbereit ist, besteht darin, die Amazon Keyspaces-system\_schema\_mcskeyspacesTabelle (system\_schema\_mcs.\*) abzufragen. Eine Liste der DDL-Anweisungen für Keyspaces finden Sie im [the section called "Keyspaces"](#) Abschnitt in der CQL-Sprachreferenz.

Die folgende Abfrage zeigt, ob ein Keyspace erfolgreich erstellt wurde.

```
SELECT * FROM system_schema_mcs.keyspaces WHERE keyspace_name = 'mykeyspace';
```

Für einen erfolgreich erstellten Keyspace sieht die Ausgabe der Abfrage wie folgt aus.

```
keyspace_name | durable_writes | replication
-----+-----+-----
```

```
mykeyspace | true | {...} 1 item
```

## Arbeiten mit Tabellen in Amazon Keyspaces

Dieser Abschnitt enthält Details zum Arbeiten mit Tabellen in Amazon Keyspaces (für Apache Cassandra).

### Themen

- [Erstellen von Tabellen in Amazon Keyspaces](#)
- [Arbeiten mit multiregionalen Tabellen in Amazon Keyspaces](#)
- [Statische Spalten in Amazon Keyspaces](#)

## Erstellen von Tabellen in Amazon Keyspaces

Amazon Keyspaces führt asynchron Data Definition Language (DDL)-Operationen wie das Erstellen und Löschen von Tabellen durch. Sie können den Erstellungsstatus neuer Tabellen in der überwachten AWS Management Console, was angibt, wann eine Tabelle aussteht oder aktiv ist. Sie können den Erstellungsstatus einer neuen Tabelle auch programmgesteuert überwachen, indem Sie die Systemschematabelle verwenden.

Eine Tabelle zeigt als aktiv im Systemschema, wenn es einsatzbereit ist. Das empfohlene Designmuster, das überprüft werden muss, wann eine neue Tabelle einsatzbereit ist, besteht darin, die Systemschematabellen von Amazon Keyspaces abzufragen (`system_schema_mcs.*`). Eine Liste der DDL-Anweisungen für Tabellen finden Sie im [the section called "Tabellen"](#) Abschnitt in der CQL-Sprachreferenz.

Die folgende Abfrage zeigt den Status einer Tabelle an.

```
SELECT keyspace_name, table_name, status FROM system_schema_mcs.tables WHERE
keyspace_name = 'mykeyspace' AND table_name = 'mytable';
```

Für eine Tabelle, die noch erstellt wird und aussteht, sieht die Ausgabe der Abfrage wie folgt aus.



```
keyspace_name | table_name | status
-----+-----+-----
mykeyspace | mytable | CREATING
```

Für eine Tabelle, die erfolgreich erstellt wurde und aktiv ist, sieht die Ausgabe der Abfrage wie folgt aus.

```
keyspace_name | table_name | status
-----+-----+-----
mykeyspace | mytable | ACTIVE
```

## Arbeiten mit multiregionalen Tabellen in Amazon Keyspaces

Für eine Tabelle mit mehreren Regionen muss die Schreibdurchsatzkapazität auf zwei Arten konfiguriert sein:

- On-Demand-Kapazitätsmodus, gemessen in Schreibenforderungseinheiten (WRUs)
- Modus bereitgestellter Kapazität mit automatischer Skalierung, gemessen in Schreibkapazitätseinheiten (WCUs)

Sie können den Modus bereitgestellter Kapazität mit Auto Scaling oder On-Demand-Kapazitätsmodus verwenden, um sicherzustellen, dass eine Tabelle mit mehreren Regionen über genügend Kapazität verfügt, um replizierte Schreibvorgänge für alle durchzuführen AWS-Regionen.

### Note

Wenn Sie den Kapazitätsmodus der Tabelle in einer der Regionen ändern, wird der Kapazitätsmodus für alle Replikate geändert.

Standardmäßig verwendet Amazon Keyspaces den On-Demand-Modus für Tabellen mit mehreren Regionen. Im On-Demand-Modus müssen Sie nicht angeben, wie viel Lese- und Schreibdurchsatz Ihre Anwendung voraussichtlich ausführen wird. Amazon Keyspaces passt sich sofort an Ihre Workloads an, wenn sie auf ein zuvor erreichtes Datenverkehrsniveau ansteigen oder sinken. Wenn der Datenverkehr eines Workloads einen neuen Höchststand erreicht, passt sich Amazon Keyspaces schnell an den Workload an.

Wenn Sie den Modus bereitgestellter Kapazität für eine Tabelle wählen, müssen Sie die Anzahl der Lesekapazitätseinheiten (RCUs) und Schreibkapazitätseinheiten (WCUs) pro Sekunde konfigurieren, die Ihre Anwendung benötigt.


Um den Durchsatzkapazitätsbedarf einer multiregionalen Tabelle zu planen, sollten Sie zunächst die Anzahl der WCUs pro Sekunde schätzen, die für jede Region benötigt werden. Anschließend fügen Sie die Schreibvorgänge aus allen Regionen hinzu, in denen Ihre Tabelle repliziert wird, und verwenden die Summe, um Kapazität für jede Region bereitzustellen. Dies ist erforderlich, da jeder Schreibvorgang, der in einer Region ausgeführt wird, auch in jeder Replikatregion wiederholt werden muss.

Wenn die Tabelle nicht über genügend Kapazität verfügt, um die Schreibvorgänge aus allen Regionen zu verarbeiten, treten Kapazitätsausnahmen auf. Darüber hinaus werden die Wartezeiten für die interregionale Replikation steigen.

Wenn Sie beispielsweise eine multiregionale Tabelle haben, in der Sie 5 Schreibvorgänge pro Sekunde in USA Ost (Nord-Virginia), 10 Schreibvorgänge pro Sekunde in USA Ost (Ohio) und 5 Schreibvorgänge pro Sekunde in Europa (Irland) erwarten, sollten Sie davon ausgehen, dass die Tabelle in jeder Region 20 WCUs verbraucht: USA Ost (Nord-Virginia), USA Ost (Ohio) und Europa (Irland). Das bedeutet, dass Sie in diesem Beispiel 20 WCUs für jedes Replikat der Tabelle bereitstellen müssen. Sie können den Kapazitätsverbrauch Ihrer Tabelle mit Amazon überwachen CloudWatch. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

Da jeder multiregionale Schreibvorgang als 1,25-fache WCUs abgerechnet wird, würden Sie in diesem Beispiel insgesamt 75 WCUs abgerechnet sehen. Weitere Informationen zu Preisen finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\) – Preise](#).

Weitere Informationen zur bereitgestellten Kapazität mit Amazon Keyspaces Auto Scaling finden Sie unter [the section called “Verwaltung der Durchsatzkapazität mit Auto Scaling”](#).

 Note

Wenn eine Tabelle im Modus bereitgestellter Kapazität mit Auto Scaling ausgeführt wird, kann die bereitgestellte Schreibkapazität innerhalb dieser Auto-Scaling-Einstellungen für jede Region schwanken.

## Statische Spalten in Amazon Keyspaces

In einer Amazon Keyspaces-Tabelle mit Clustering-Spalten können Sie das `STATIC` Schlüsselwort verwenden, um eine statische Spalte zu erstellen. Der in einer statischen Spalte gespeicherte Wert wird von allen Zeilen in einer logischen Partition gemeinsam genutzt. Wenn Sie den Wert dieser Spalte aktualisieren, wendet Amazon Keyspaces die Änderung automatisch auf alle Zeilen in der Partition an.

In diesem Abschnitt wird beschrieben, wie Sie die codierte Datengröße berechnen, wenn Sie in statische Spalten schreiben. Dieser Prozess wird getrennt von dem Prozess behandelt, der Daten in die nichtstatischen Spalten einer Zeile schreibt. Zusätzlich zu den Größenkontingenten für statische Daten wirken sich Lese- und Schreibvorgänge für statische Spalten auch unabhängig auf die Messungs- und Durchsatzkapazität für Tabellen aus.

### Berechnen der statischen Spaltengröße pro logischer Partition in Amazon Keyspaces

Dieser Abschnitt enthält Details zur Schätzung der codierten Größe statischer Spalten in Amazon Keyspaces. Die codierte Größe wird verwendet, wenn Sie Ihre -Rechnung und die Kontingentnutzung berechnen. Sie sollten auch die codierte Größe verwenden, wenn Sie die Kapazitätsanforderungen für den bereitgestellten Durchsatz für Tabellen berechnen. Um die codierte Größe statischer Spalten in Amazon Keyspaces zu berechnen, können Sie die folgenden Richtlinien verwenden.

- Partitionsschlüssel können bis zu 2048 Byte an Daten enthalten. Jede Spalte im Partitionsschlüssel benötigt bis zu 3 Byte Metadaten. Diese Metadatenbytes werden auf Ihr Kontingent für statische Datengrößen von 1 MB pro Partition angerechnet. Bei der Berechnung der Größe Ihrer statischen Daten sollten Sie davon ausgehen, dass jede Partitionsschlüsselspalte die vollständigen 3 Byte Metadaten verwendet.
- Verwenden Sie die Rohgröße der statischen Spaltendatenwerte basierend auf dem Datentyp. Weitere Informationen zu Datentypen finden Sie unter [the section called “Datentypen”](#).
- Fügen Sie der Größe der statischen Daten für Metadaten 104 Byte hinzu.
- Clustering-Spalten und reguläre, nicht primäre Spalten werden nicht auf die Größe statischer Daten angerechnet. Informationen zum Schätzen der Größe nichtstatischer Daten innerhalb von Zeilen finden Sie unter [the section called “Berechnung der Zeilengröße”](#).

Die gesamte codierte Größe einer statischen Spalte basiert auf der folgenden Formel:

```
partition key columns + static columns + metadata = total encoded size of static data
```

Betrachten Sie das folgende Beispiel für eine Tabelle, in der alle Spalten vom Typ Ganzzahl sind. Die Tabelle hat zwei Partitionsschlüsselspalten, zwei Clustering-Spalten, eine reguläre Spalte und eine statische Spalte.

```
CREATE TABLE mykeyspace.mytable(pk_col1 int, pk_col2 int, ck_col1 int, ck_col2
int, reg_col1 int, static_col1 int static, primary key((pk_col1, pk_col2),ck_col1,
ck_col2));
```

In diesem Beispiel berechnen wir die Größe der statischen Daten der folgenden Anweisung:

```
INSERT INTO mykeyspace.mytable (pk_col1, pk_col2, static_col1) values(1,2,6);
```

Um die Gesamtzahl der Bytes zu schätzen, die für diesen Schreibvorgang erforderlich sind, können Sie die folgenden Schritte ausführen.

1. Berechnen Sie die Größe einer Partitionsschlüsselspalte, indem Sie die Bytes für den in der Spalte gespeicherten Datentyp und die Metadatenbytes hinzufügen. Wiederholen Sie dies für alle Partitionsschlüsselspalten.

- a. Berechnen Sie die Größe der ersten Spalte des Partitionsschlüssels (pk\_col1):

```
4 bytes for the integer data type + 3 bytes for partition key metadata = 7
bytes
```

- b. Berechnen Sie die Größe der zweiten Spalte des Partitionsschlüssels (pk\_col2):

```
4 bytes for the integer data type + 3 bytes for partition key metadata = 7
bytes
```

- c. Fügen Sie beide Spalten hinzu, um die geschätzte Gesamtgröße der Partitionsschlüsselspalten zu erhalten:

```
7 bytes + 7 bytes = 14 bytes for the partition key columns
```

2. Fügen Sie die Größe der statischen Spalten hinzu. In diesem Beispiel haben wir nur eine statische Spalte, die eine Ganzzahl speichert (die 4 Byte erfordert).
3. Um die gesamte codierte Größe der statischen Spaltendaten zu erhalten, addieren Sie schließlich die Bytes für die Primärschlüsselspalten und statischen Spalten und fügen Sie die zusätzlichen 104 Bytes für Metadaten hinzu:

```
14 bytes for the partition key columns + 4 bytes for the static column + 104 bytes
for metadata = 122 bytes.
```

Sie können statische und nichtstatische Daten auch mit derselben Anweisung aktualisieren. Um die Gesamtgröße des Schreibvorgangs zu schätzen, müssen Sie zunächst die Größe der nichtstatischen Datenaktualisierung berechnen. Berechnen Sie dann die Größe der Zeilenaktualisierung, wie im Beispiel unter gezeigt [the section called “Berechnung der Zeilengröße”](#), und fügen Sie die Ergebnisse hinzu.

In diesem Fall können Sie insgesamt 2 MB bis 1 MB das maximale Kontingent für Zeilengrößen und 1 MB das Kontingent für die maximale statische Datengröße pro logischer Partition schreiben.

Um die Gesamtgröße einer Aktualisierung statischer und nichtstatischer Daten in derselben Anweisung zu berechnen, können Sie die folgende Formel verwenden:

```
(partition key columns + static columns + metadata = total encoded size of static data)
+ (partition key columns + clustering columns + regular columns + row metadata = total
encoded size of row)
= total encoded size of data written
```

Betrachten Sie das folgende Beispiel für eine Tabelle, in der alle Spalten vom Typ Ganzzahl sind. Die Tabelle hat zwei Partitionsschlüsselspalten, zwei Clustering-Spalten, eine reguläre Spalte und eine statische Spalte.

```
CREATE TABLE mykeyspace.mytable(pk_col1 int, pk_col2 int, ck_col1 int, ck_col2
int, reg_col1 int, static_col1 int static, primary key((pk_col1, pk_col2),ck_col1,
ck_col2));
```

In diesem Beispiel berechnen wir die Datengröße, wenn wir eine Zeile in die Tabelle schreiben, wie in der folgenden Anweisung gezeigt:

```
INSERT INTO mykeyspace.mytable (pk_col1, pk_col2, ck_col1, ck_col2, reg_col1,
static_col1) values(2,3,4,5,6,7);
```

Um die Gesamtzahl der Bytes zu schätzen, die für diesen Schreibvorgang erforderlich sind, können Sie die folgenden Schritte ausführen.

1. Berechnen Sie die codierte Gesamtgröße statischer Daten wie oben gezeigt. In diesem Beispiel sind es 122 Byte.
2. Fügen Sie die Größe der gesamten codierten Größe der Zeile basierend auf der Aktualisierung nichtstatischer Daten hinzu, indem Sie die Schritte unter befolgen [the section called "Berechnung der Zeilengröße"](#). In diesem Beispiel beträgt die Gesamtgröße der Zeilenaktualisierung 134 Byte.

```
122 bytes for static data + 134 bytes for nonstatic data = 256 bytes.
```

## Messung von Lese-/Schreibvorgängen statischer Daten in Amazon Keyspaces

Statische Daten sind mit logischen Partitionen in Cassandra verknüpft, nicht mit einzelnen Zeilen. Logische Partitionen in Amazon Keyspaces können praktisch unbegrenzt groß sein, indem sie sich über mehrere physische Speicherpartitionen erstrecken. Daher misst Amazon Keyspaces Schreibvorgänge für statische und nichtstatische Daten separat. Darüber hinaus erfordern Schreibvorgänge, die sowohl statische als auch nichtstatische Daten enthalten, zusätzliche zugrunde liegende Operationen, um Datenkonsistenz zu gewährleisten.

Wenn Sie einen gemischten Schreibvorgang sowohl statischer als auch nichtstatischer Daten ausführen, führt dies zu zwei separaten Schreibvorgängen – eine für nichtstatische Daten und eine für statische Daten. Dies gilt sowohl für On-Demand- als auch für bereitgestellte Lese-/Schreibkapazitätsmodi.

Das folgende Beispiel enthält Details zur Schätzung der erforderlichen Lesekapazitätseinheiten (RCUs) und Schreibkapazitätseinheiten (WCUs), wenn Sie die Anforderungen an die bereitgestellte Durchsatzkapazität für Tabellen in Amazon Keyspaces berechnen, die statische Spalten haben. Sie können schätzen, wie viel Kapazität Ihre Tabelle benötigt, um Schreibvorgänge zu verarbeiten, die sowohl statische als auch nichtstatische Daten enthalten, indem Sie die folgende Formel verwenden:

```
2 x WCUs required for nonstatic data + 2 x WCUs required for static data
```

Wenn Ihre Anwendung beispielsweise 27 KBs Daten pro Sekunde schreibt und jeder Schreibvorgang 25,5 KBs nichtstatischer Daten und 1,5 KBs statischer Daten enthält, benötigt Ihre Tabelle 56 WCUs ( $2 \times 26 \text{ WCUs} + 2 \times 2 \text{ WCUs}$ ).

Amazon Keyspaces misst die Lesevorgänge statischer und nichtstatischer Daten genauso wie Lesevorgänge mehrerer Zeilen. Daher basiert der Preis für das Lesen statischer und nichtstatischer Daten in derselben Operation auf der Gesamtgröße der Daten, die zur Durchführung des Lesevorgangs verarbeitet wurden.

Informationen zur Überwachung von Serverless-Ressourcen mit Amazon CloudWatch finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

## Arbeiten mit Zeilen in Amazon Keyspaces

Dieser Abschnitt enthält Einzelheiten zur Arbeit mit Zeilen in Amazon Keyspaces (für Apache Cassandra). Tabellen sind die primären Datenstrukturen in Amazon Keyspaces, und Daten in Tabellen sind in Spalten und Zeilen organisiert.

Themen

- [Berechnung der Zeilengröße in Amazon Keyspaces](#)

### Berechnung der Zeilengröße in Amazon Keyspaces

Amazon Keyspaces bietet vollständig verwalteten Speicher, der eine Lese- und Schreibleistung im einstelligen Millisekundenbereich bietet und Daten dauerhaft in mehreren Availability Zones speichert. AWS Amazon Keyspaces fügt Metadaten an alle Zeilen und Primärschlüsselspalten an, um einen effizienten Datenzugriff und hohe Verfügbarkeit zu unterstützen.

Dieser Abschnitt enthält Einzelheiten zur Schätzung der codierten Zeilengröße in Amazon Keyspaces. Die kodierte Zeilengröße wird bei der Berechnung Ihrer Rechnung und der Kontingentnutzung verwendet. Sie sollten auch die kodierte Zeilengröße verwenden, wenn Sie die Anforderungen an die bereitgestellte Durchsatzkapazität für Tabellen berechnen. Um die kodierte Größe von Zeilen in Amazon Keyspaces zu berechnen, können Sie die folgenden Richtlinien verwenden.

- Verwenden Sie für reguläre Spalten, bei denen es sich nicht um Primärschlüssel, Clusterspalten oder STATIC Spalten handelt, die Rohgröße der Zellen basierend auf dem Datentyp und fügen Sie die erforderlichen Metadaten hinzu. Weitere Informationen zu den in Amazon Keyspaces unterstützten Datentypen finden Sie unter [the section called “Datentypen”](#). Einige wichtige Unterschiede in der Art und Weise, wie Amazon Keyspaces Datentypwerte und Metadaten speichert, sind unten aufgeführt.
- Der für jeden Spaltennamen benötigte Speicherplatz wird unter Verwendung einer Spalten-ID gespeichert und zu jedem in der Spalte gespeicherten Datenwert hinzugefügt. Der Speicherwert der Spalten-ID hängt von der Gesamtzahl der Spalten in Ihrer Tabelle ab:
  - 1—62 Spalten: 1 Byte

- 63—124 Spalten: 2 Byte
- 125—186 Spalten: 3 Byte

Fügen Sie für jede weitere 62 Spalten 1 Byte hinzu. Beachten Sie, dass in Amazon Keyspaces bis zu 225 reguläre Spalten mit einer einzigen INSERT UPDATE OR-Anweisung geändert werden können. Weitere Informationen finden Sie unter [the section called “Service Quotas für Amazon Keyspaces”](#).

- Partitionsschlüssel können bis zu 2048 Byte an Daten enthalten. Jede Schlüsselspalte im Partitionsschlüssel benötigt bis zu 3 Byte an Metadaten. Bei der Berechnung der Größe Ihrer Zeile sollten Sie davon ausgehen, dass jede Partitionsschlüsselspalte die vollen 3 Byte an Metadaten verwendet.
- In Clusterspalten können bis zu 850 Byte an Daten gespeichert werden. Zusätzlich zur Größe des Datenwerts benötigt jede Clusterspalte bis zu 20% der Datenwertgröße für Metadaten. Bei der Berechnung der Größe Ihrer Zeile sollten Sie 1 Byte Metadaten für jeweils 5 Byte des Datenwerts der Clusterspalte hinzufügen.
- Amazon Keyspaces speichert den Datenwert jeder Partitionsschlüssel- und Clusterschlüsselspalte zweimal. Der zusätzliche Overhead wird für effizientes Abfragen und integrierte Indizierung verwendet.
- Cassandra- ASCIITEXT, und VARCHAR String-Datentypen werden alle in Amazon Keyspaces unter Verwendung von Unicode mit UTF-8-Binärcodierung gespeichert. Die Größe einer Zeichenfolge in Amazon Keyspaces entspricht der Anzahl der UTF-8-kodierten Byte.
- CassandraINT, BIGINTSMALLINT, und TINYINT Datentypen werden in Amazon Keyspaces als Datenwerte mit variabler Länge mit bis zu 38 signifikanten Ziffern gespeichert. Nullen am Anfang und am Ende werden abgeschnitten. Die Größe jedes dieser Datentypen beträgt ungefähr 1 Byte pro zwei signifikanten Ziffern + 1 Byte.
- A BLOB in Amazon Keyspaces wird mit der Roh-Bytelänge des Werts gespeichert.
- Die Größe eines Null Werts oder Boolean Werts beträgt 1 Byte.
- Eine Spalte, die unabhängig von ihrem Inhalt Sammlungsdatentypen wie Metadaten speichert LIST oder 3 Byte an Metadaten MAP benötigt. Die Größe eines LIST oder MAP ist (Spalten-ID) + Summe (Größe der verschachtelten Elemente) + (3 Byte). Die Größe eines leeren LIST Or MAP ist (Spalten-ID) + (3 Byte). Jedes Individuum LIST oder MAP Element benötigt außerdem 1 Byte an Metadaten.
- STATICSpaltendaten werden nicht auf die maximale Zeilengröße von 1 MB angerechnet. Informationen zur Berechnung der Datengröße statischer Spalten finden Sie unter [the section called “Berechnen der statischen Spaltengröße pro logischer Partition”](#).



- Wenn die Funktion aktiviert ist, werden clientseitige Zeitstempel für jede Spalte in jeder Zeile gespeichert. Diese Zeitstempel nehmen ungefähr 20–40 Byte in Anspruch (abhängig von Ihren Daten) und tragen zu den Speicher- und Durchsatzkosten für die Zeile bei. Weitere Informationen finden Sie unter [the section called “Clientseitige Zeitstempel in Amazon Keyspaces”](#).
- Fügen Sie der Größe jeder Zeile für Zeilenmetadaten 100 Byte hinzu.

Die Gesamtgröße einer codierten Datenzeile basiert auf der folgenden Formel:

```
partition key columns + clustering columns + regular columns + row metadata = total encoded size of row
```

#### Important

Alle Spaltenmetadaten, z. B. Spalten-IDs, Partitionsschlüssel-Metadaten, Cluster-Spaltenmetadaten sowie clientseitige Zeitstempel und Zeilenmetadaten werden auf die maximale Zeilengröße von 1 MB angerechnet.

Betrachten Sie das folgende Beispiel für eine Tabelle, in der alle Spalten vom Typ Integer sind. Die Tabelle hat zwei Partitionsschlüsselspalten, zwei Clusterspalten und eine reguläre Spalte. Da diese Tabelle fünf Spalten hat, beträgt der für den Spaltennamen benötigte Speicherplatz 1 Byte.

```
CREATE TABLE mykeyspace.mytable(pk_col1 int, pk_col2 int, ck_col1 int, ck_col2 int, reg_col1 int, primary key((pk_col1, pk_col2),ck_col1, ck_col2));
```

In diesem Beispiel berechnen wir die Datengröße, wenn wir eine Zeile in die Tabelle schreiben, wie in der folgenden Anweisung dargestellt:

```
INSERT INTO mykeyspace.mytable (pk_col1, pk_col2, ck_col1, ck_col2, reg_col1) values(1,2,3,4,5);
```

Um die Gesamtanzahl der für diesen Schreibvorgang benötigten Byte zu schätzen, können Sie die folgenden Schritte verwenden.

1. Berechnen Sie die Größe einer Partitionsschlüsselspalte, indem Sie die Byte für den in der Spalte gespeicherten Datentyp und die Metadaten-Bytes addieren. Wiederholen Sie diesen Vorgang für alle Partitionsschlüsselspalten.

- a. Berechnet die Größe der ersten Spalte des Partitionsschlüssels (pk\_col1):

```
(2 bytes for the integer data type) x 2 + 1 byte for the column id + 3 bytes
for partition key metadata = 8 bytes
```

- b. Berechnet die Größe der zweiten Spalte des Partitionsschlüssels (pk\_col2):

```
(2 bytes for the integer data type) x 2 + 1 byte for the column id + 3 bytes
for partition key metadata = 8 bytes
```

- c. Fügen Sie beide Spalten hinzu, um die geschätzte Gesamtgröße der Partitionsschlüsselspalten zu erhalten:

```
8 bytes + 8 bytes = 16 bytes for the partition key columns
```

2. Berechnen Sie die Größe der Clusterspalte, indem Sie die Byte für den in der Spalte gespeicherten Datentyp und die Metadaten-Bytes hinzufügen. Wiederholen Sie diesen Vorgang für alle Clusterspalten.

- a. Berechnet die Größe der ersten Spalte der Cluster-Spalte (ck\_col1):

```
(2 bytes for the integer data type) x 2 + 20% of the data value (2 bytes) for
clustering column metadata + 1 byte for the column id = 6 bytes
```

- b. Berechnet die Größe der zweiten Spalte der Cluster-Spalte (ck\_col2):

```
(2 bytes for the integer data type) x 2 + 20% of the data value (2 bytes) for
clustering column metadata + 1 byte for the column id = 6 bytes
```

- c. Fügen Sie beide Spalten hinzu, um die geschätzte Gesamtgröße der Cluster-Spalten zu erhalten:

```
6 bytes + 6 bytes = 12 bytes for the clustering columns
```

3. Fügen Sie die Größe der regulären Spalten hinzu. In diesem Beispiel haben wir nur eine Spalte, die eine einstellige Ganzzahl speichert, was 2 Byte mit 1 Byte für die Spalten-ID erfordert.
4. Um schließlich die gesamte kodierte Zeilengröße zu ermitteln, addieren Sie die Byte für alle Spalten und fügen Sie die zusätzlichen 100 Byte für die Zeilenmetadaten hinzu:

16 bytes for the partition key columns + 12 bytes for clustering columns + 3 bytes for the regular column + 100 bytes for row metadata = 131 bytes.

Informationen zur Überwachung serverloser Ressourcen mit Amazon finden Sie CloudWatch unter [the section called “Überwachung mit CloudWatch”](#).

## Arbeiten mit Abfragen in Amazon Keyspaces

Dieser Abschnitt enthält eine Einführung in die Arbeit mit Abfragen in Amazon Keyspaces (für Apache Cassandra). Die verfügbaren CQL-Anweisungen zum Abfragen, Transformieren und Verwalten von Daten lauten `SELECT`, `INSERT`, `UPDATE` und `DELETE`. In den folgenden Themen werden einige der komplexeren Optionen beschrieben, die bei der Arbeit mit Abfragen verfügbar sind. Die vollständige Sprachsyntax mit Beispielen finden Sie unter [the section called “DML-Anweisungen.”](#).

### Themen

- [Den IN Operator mit dem SELECT Statement in Amazon Keyspaces verwenden](#)
- [Ergebnisse in Amazon Keyspaces bestellen](#)
- [Paginierung von Ergebnissen in Amazon Keyspaces](#)

## Den **IN** Operator mit dem **SELECT** Statement in Amazon Keyspaces verwenden

### WÄHLEN SIE IN

Sie können Daten aus Tabellen mit der `SELECT` Anweisung abfragen, die eine oder mehrere Spalten für eine oder mehrere Zeilen in einer Tabelle liest und eine Ergebnismenge zurückgibt, die die Zeilen enthält, die der Anforderung entsprechen. Eine `SELECT` Anweisung enthält eine `select_clause`, die bestimmt, welche Spalten gelesen und in der Ergebnismenge zurückgegeben werden sollen. Die Klausel kann Anweisungen zum Transformieren der Daten enthalten, bevor sie zurückgegeben werden. Die optionale `WHERE` Klausel gibt an, welche Zeilen abgefragt werden müssen, und besteht aus Beziehungen zu den Spalten, die Teil des Primärschlüssels sind. Amazon Keyspaces unterstützt das `IN` Schlüsselwort in der `WHERE` Klausel. In diesem Abschnitt wird anhand von Beispielen gezeigt, wie Amazon Keyspaces `SELECT` Anweisungen mit dem `IN` Schlüsselwort verarbeitet.

Dieses Beispiel zeigt, wie Amazon Keyspaces die SELECT Anweisung mit dem IN Schlüsselwort in Unterabfragen aufteilt. In diesem Beispiel verwenden wir eine Tabelle mit dem Namen `my_keyspace.customers`. Die Tabelle hat eine Primärschlüsselspalte `department_id`, zwei Clusterspalten `sales_region_id` und eine Spalte `sales_representative_id`, die den Namen des Kunden in der `customer_name` Spalte enthält.

```
SELECT * FROM my_keyspace.customers;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 0               | 0                       | a             |
| 0             | 0               | 1                       | b             |
| 0             | 1               | 0                       | c             |
| 0             | 1               | 1                       | d             |
| 1             | 0               | 0                       | e             |
| 1             | 0               | 1                       | f             |
| 1             | 1               | 0                       | g             |
| 1             | 1               | 1                       | h             |

Mithilfe dieser Tabelle können Sie die folgende SELECT Anweisung ausführen, um anhand des IN Schlüsselworts in der WHERE Klausel die Kunden in den Abteilungen und Vertriebsregionen zu finden, an denen Sie interessiert sind. Die folgende Aussage ist ein Beispiel dafür.

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (0, 1) AND sales_region_id IN (0, 1);
```

Amazon Keyspaces unterteilt diese Anweisung in vier Unterabfragen, wie in der folgenden Ausgabe dargestellt.

```
SELECT * FROM my_keyspace.customers WHERE department_id = 0 AND sales_region_id = 0;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 0               | 0                       | a             |
| 0             | 0               | 1                       | b             |

```
SELECT * FROM my_keyspace.customers WHERE department_id = 0 AND sales_region_id = 1;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 1               | 0                       | c             |

```

0 | 1 | 1 | d
SELECT * FROM my_keyspace.customers WHERE department_id = 1 AND sales_region_id = 0;

department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
1 | 0 | 0 | e
1 | 0 | 1 | f

SELECT * FROM my_keyspace.customers WHERE department_id = 1 AND sales_region_id = 1;

department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
1 | 1 | 0 | g
1 | 1 | 1 | h

```

Wenn das `IN` Schlüsselwort verwendet wird, paginiert Amazon Keyspaces die Ergebnisse in den folgenden Fällen automatisch:

- Nachdem jede zehnte Unterabfrage verarbeitet wurde.
- Nach der Verarbeitung von 1 MB logischer I/O.
- Wenn Sie eine konfiguriert haben `PAGE SIZE`, paginiert Amazon Keyspaces nach dem Lesen der Anzahl der Abfragen, die auf der Gruppe basieren, die verarbeitet werden sollen. `PAGE SIZE`
- Wenn Sie das `LIMIT` Schlüsselwort verwenden, um die Anzahl der zurückgegebenen Zeilen zu reduzieren, paginiert Amazon Keyspaces, nachdem es die Anzahl der Abfragen gelesen hat, die basierend auf dem Satz verarbeitet werden sollen. `LIMIT`

Die folgende Tabelle wird verwendet, um dies anhand eines Beispiels zu veranschaulichen.

Weitere Hinweise zur Seitennummerierung finden Sie unter [the section called “Ergebnisse paginieren”](#).

```

SELECT * FROM my_keyspace.customers;

department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
2 | 0 | 0 | g
2 | 1 | 1 | h
2 | 2 | 2 | i
0 | 0 | 0 | a

```

|   |  |   |  |   |  |   |
|---|--|---|--|---|--|---|
| 0 |  | 1 |  | 1 |  | b |
| 0 |  | 2 |  | 2 |  | c |
| 1 |  | 0 |  | 0 |  | d |
| 1 |  | 1 |  | 1 |  | e |
| 1 |  | 2 |  | 2 |  | f |
| 3 |  | 0 |  | 0 |  | j |
| 3 |  | 1 |  | 1 |  | k |
| 3 |  | 2 |  | 2 |  | l |

Sie können die folgende Anweisung für diese Tabelle ausführen, um zu sehen, wie die Paginierung funktioniert.

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (0, 1, 2, 3) AND
sales_region_id IN (0, 1, 2) AND sales_representative_id IN (0, 1);
```

Amazon Keyspaces verarbeitet diese Anweisung als 24 Unterabfragen, da die Kardinalität des kartesischen Produkts aller in dieser Abfrage enthaltenen IN Begriffe 24 ist.

```
department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
0 | 0 | 0 | a
0 | 1 | 1 | b
1 | 0 | 0 | d
1 | 1 | 1 | e

---MORE---
department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
2 | 0 | 0 | g
2 | 1 | 1 | h
3 | 0 | 0 | j

---MORE---
department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
3 | 1 | 1 | k
```

Dieses Beispiel zeigt, wie Sie die ORDER BY Klausel in einer Anweisung mit dem Schlüsselwort verwenden können. SELECT IN

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (3, 2, 1) ORDER BY
sales_region_id DESC;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 3             | 2               | 2                       | l             |
| 3             | 1               | 1                       | k             |
| 3             | 0               | 0                       | j             |
| 2             | 2               | 2                       | i             |
| 2             | 1               | 1                       | h             |
| 2             | 0               | 0                       | g             |
| 1             | 2               | 2                       | f             |
| 1             | 1               | 1                       | e             |
| 1             | 0               | 0                       | d             |

Unterabfragen werden in der Reihenfolge verarbeitet, in der die Spalten Partitionsschlüssel und Clusterschlüssel in der Abfrage dargestellt werden. Im folgenden Beispiel werden Unterabfragen für den Partitionsschlüsselwert“ 2 „zuerst verarbeitet, gefolgt von Unterabfragen für den Partitionsschlüsselwert“ 3 „und“ 1“. Die Ergebnisse einer bestimmten Unterabfrage werden gemäß der Sortierklausel der Abfrage, falls vorhanden, oder der bei der Tabellenerstellung definierten Clusterreihenfolge der Tabelle sortiert.

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (2, 3, 1) ORDER BY
sales_region_id DESC;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 2             | 2               | 2                       | i             |
| 2             | 1               | 1                       | h             |
| 2             | 0               | 0                       | g             |
| 3             | 2               | 2                       | l             |
| 3             | 1               | 1                       | k             |
| 3             | 0               | 0                       | j             |
| 1             | 2               | 2                       | f             |
| 1             | 1               | 1                       | e             |
| 1             | 0               | 0                       | d             |

## Ergebnisse in Amazon Keyspaces bestellen

Die `ORDER BY` Klausel gibt die Sortierreihenfolge der in einer `SELECT` Anweisung zurückgegebenen Ergebnisse an. Die Anweisung verwendet eine Liste von Spaltennamen als Argumente, und für jede Spalte können Sie die Sortierreihenfolge für die Daten angeben. Sie können Gruppierungsspalten nur in Sortierklauseln angeben. Spalten, die keine Gruppierung bilden, sind nicht zulässig.

Die beiden verfügbaren Sortierreihenoptionen für die zurückgegebenen Ergebnisse sind für die aufsteigende und ASC DESC für die absteigende Sortierreihenfolge.

```
SELECT * FROM my_keyspace.my_table ORDER BY (col1 ASC, col2 DESC, col3 ASC);
```

| col1 | col2 | col3 |
|------|------|------|
| 0    | 6    | a    |
| 1    | 5    | b    |
| 2    | 4    | c    |
| 3    | 3    | d    |
| 4    | 2    | e    |
| 5    | 1    | f    |
| 6    | 0    | g    |

```
SELECT * FROM my_keyspace.my_table ORDER BY (col1 DESC, col2 ASC, col3 DESC);
```

| col1 | col2 | col3 |
|------|------|------|
| 6    | 0    | g    |
| 5    | 1    | f    |
| 4    | 2    | e    |
| 3    | 3    | d    |
| 2    | 4    | c    |
| 1    | 5    | b    |
| 0    | 6    | a    |

Wenn Sie die Sortierreihenfolge in der Abfrageanweisung nicht angeben, wird die Standardreihenfolge der Clusterspalte verwendet.

Die möglichen Sortierreihenfolgen, die Sie in einer Sortierklausel verwenden können, hängen von der Sortierreihenfolge ab, die jeder Clusterspalte bei der Tabellenerstellung zugewiesen wurde. Abfrageergebnisse können nur in der Reihenfolge sortiert werden, die bei der Tabellenerstellung für alle Clusterspalten definiert wurde, oder in der umgekehrten Reihenfolge der definierten Sortierreihenfolge. Andere mögliche Kombinationen sind nicht zulässig.

Wenn die Tabelle beispielsweise (col1 ASC, col2 DESC, col3 ASC) CLUSTERING ORDER lautet, dann ORDER BY sind die gültigen Parameter für entweder (col1 ASC, col2 DESC, col3 ASC) oder (col1 DESC, col2 ASC, col3 DESC) gültig. Weitere Informationen zu finden Sie unter. CLUSTERING ORDER table\_options [the section called "CREATE TABLE"](#)



## Paginierung von Ergebnissen in Amazon Keyspaces

Amazon Keyspaces paginiert die Ergebnisse von SELECT Kontoauszügen automatisch, wenn die zur Verarbeitung der SELECT Anweisung gelesenen Daten 1 MB überschreiten. Bei der Paginierung werden die SELECT Abrechnungsergebnisse in „Seiten“ mit Daten aufgeteilt, die 1 MB (oder weniger) groß sind. Eine Anwendung kann die erste Ergebnisseite verarbeiten, dann die zweite Seite und so weiter. Kunden sollten bei der Verarbeitung von SELECT Abfragen, die mehrere Zeilen zurückgeben, immer nach Paginierungstoken suchen.

Wenn ein Kunde eine liefert, für PAGE SIZE die mehr als 1 MB an Daten gelesen werden müssen, teilt Amazon Keyspaces die Ergebnisse automatisch auf der Grundlage der 1-MB-Datenlesegeschritte in mehrere Seiten auf.

Wenn die durchschnittliche Größe einer Zeile beispielsweise 100 KB beträgt und Sie einen Wert PAGE SIZE von 20 angeben, paginiert Amazon Keyspaces Daten automatisch, nachdem es 10 Zeilen gelesen hat (1000 KB gelesener Daten).

Da Amazon Keyspaces Ergebnisse auf der Grundlage der Anzahl der Zeilen paginiert, die es zur Bearbeitung einer Anfrage liest, und nicht auf der Anzahl der im Ergebnissatz zurückgegebenen Zeilen, enthalten einige Seiten möglicherweise keine Zeilen, wenn Sie gefilterte Abfragen ausführen.

Wenn Sie beispielsweise PAGE SIZE auf 10 setzen und Keyspaces 30 Zeilen auswertet, um Ihre SELECT Abfrage zu verarbeiten, gibt Amazon Keyspaces drei Seiten zurück. Wenn nur eine Teilmenge der Zeilen mit Ihrer Anfrage übereinstimmt, haben einige Seiten möglicherweise weniger als 10 Zeilen. Ein Beispiel dafür, wie sich LIMIT Abfragen auf PAGE SIZE die Lesekapazität auswirken können, finden Sie unter [the section called “Abfragen einschränken”](#).

## Arbeiten mit Partitionierern in Amazon Keyspaces

In Apache Cassandra steuern Partitionierer, auf welchen Knoten Daten im Cluster gespeichert werden. Partitionierer erstellen ein numerisches Token unter Verwendung eines Hashwerts des Partitionsschlüssels. Cassandra verwendet dieses Token, um Daten auf Knoten zu verteilen. Kunden können diese Token auch in SELECT Operationen und WHERE Klauseln verwenden, um Lese- und Schreibvorgänge zu optimieren. Beispielsweise können Clients effizient parallel Abfragen an großen Tabellen durchführen, indem sie in jedem parallelen Job unterschiedliche Tokenbereiche angeben, die abgefragt werden sollen.

Amazon Keyspaces bietet drei verschiedene Partitionierer.

## Murmur3 Partitioner (Standard)

Apache Cassandra-kompatibel `Murmur3Partitioner`. Das `Murmur3Partitioner` ist die Standard-Cassandra-Partitionierung in Amazon Keyspaces und in Cassandra 1.2 und späteren Versionen.

## RandomPartitioner

Apache Cassandra-kompatibel. Das `RandomPartitioner` ist die Standard-Cassandra-Partitionierung für Versionen vor Cassandra 1.2.

## Keyspaces Standardpartitionierung

Der `DefaultPartitioner` gibt dieselben token Funktionsergebnisse zurück wie der `RandomPartitioner`.

Die Partitionierungseinstellung wird pro Region auf Kontoebene angewendet. Wenn Sie beispielsweise den Partitionierer in USA Ost (Nord-Virginia) ändern, wird die Änderung auf alle Tabellen desselben Kontos in dieser Region angewendet. Sie können Ihre Partitionierung jederzeit sicher ändern. Beachten Sie, dass es ungefähr 10 Minuten dauert, bis die Konfigurationsänderung abgeschlossen ist. Sie müssen Ihre Amazon Keyspaces-Daten nicht neu laden, wenn Sie die Partitionierungseinstellung ändern. Clients verwenden automatisch die neue Partitionierungseinstellung, wenn sie das nächste Mal eine Verbindung herstellen.

Sie können die Partitionierung ändern, indem Sie die AWS Management Console oder Cassandra Query Language (CQL) verwenden.

## AWS Management Console

So ändern Sie den Partitionierer mithilfe der Amazon Keyspaces-Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Configuration (Konfiguration) aus.
3. Gehen Sie auf der Konfigurationsseite zu Partitionierung bearbeiten.
4. Wählen Sie den Partitionierer aus, der mit Ihrer Version von Cassandra kompatibel ist. Es dauert ungefähr 10 Minuten, bis die Änderung des Partitionierers angewendet wird.

**Note**

Nachdem die Konfigurationsänderung abgeschlossen ist, müssen Sie die Verbindung zu Amazon Keyspaces trennen und erneut verbinden, um Anfragen zur Verwendung der neuen Partitionierung zu stellen.

## Cassandra Query Language (CQL)

1. Um zu sehen, welcher Partitionierer für das Konto konfiguriert ist, können Sie die folgende Abfrage verwenden.

```
SELECT partitioner from system.local;
```

Wenn der Partitionierer nicht geändert wurde, hat die Abfrage die folgende Ausgabe.

```
partitioner

com.amazonaws.cassandra.DefaultPartitioner
```

2. Um die Partitionierung auf die Partitionierung zu aktualisieren, können Sie die Murmur3 folgende Anweisung verwenden.

```
UPDATE system.local set
partitioner='org.apache.cassandra.dht.Murmur3Partitioner' where key='local';
```

3. Beachten Sie, dass es ungefähr 10 Minuten dauert, bis diese Konfigurationsänderung abgeschlossen ist. Um zu bestätigen, dass der Partitionierer eingerichtet wurde, können Sie die SELECT Abfrage erneut ausführen. Beachten Sie, dass die Antwort aufgrund der eventuellen Lesekonsistenz möglicherweise noch nicht die Ergebnisse der kürzlich abgeschlossenen Änderung der Partitionierung widerspiegelt. Wenn Sie den SELECT Vorgang nach kurzer Zeit erneut wiederholen, sollte die Antwort die neuesten Daten zurückgeben.

```
SELECT partitioner from system.local;
```

**Note**

Sie müssen die Verbindung zu Amazon Keyspaces trennen und erneut verbinden, damit Anfragen den neuen Partitionierer verwenden.

## Arbeiten mit Tags und Labels für Amazon Keyspaces-Ressourcen

Sie können Amazon Keyspaces (für Apache Cassandra)-Ressourcen mit Tags kennzeichnen. Mit Tags können Sie Ihre Ressourcen auf unterschiedliche Weise kategorisieren, z. B. nach Zweck, Eigentümer, Umgebung oder anderen Kriterien. Tags können Sie bei Folgendem unterstützen:

- Eine Ressource basierend auf den Tags, die Sie ihr zugeordnet haben, schnell zu erkennen.
- Siehe AWS Rechnungen, aufgeschlüsselt nach Tags.
- Steuern Sie den Zugriff auf Amazon Keyspaces-Ressourcen basierend auf Tags. Beispiele für IAM-Richtlinien, die Tags verwenden, finden Sie unter [the section called “Autorisierung basierend auf Amazon Keyspaces-Tags”](#).

Tagging wird von - AWS Services wie Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon Keyspaces und mehr unterstützt. Effizientes Tagging kann Kosteneinblicke bereistellen, mit denen Sie Berichte für Services erstellen können, die ein spezifisches Tag aufweisen.

Gehen Sie wie folgt vor, um sich mit dem Taggen vertraut zu machen::

1. Lesen Sie [Tagging-Einschränkungen für Amazon Keyspaces](#).
2. Tags mit [von Operationen für Amazon Keyspaces](#) erstellen.
3. Verwenden Sie [Kostenzuordnungsberichte für Amazon Keyspaces](#), um Ihre AWS Kosten pro aktivem Tag zu verfolgen.

Schließlich wird empfohlen, optimale Tagging-Strategien zu befolgen. Weitere Informationen finden Sie unter [AWS -Markierungsstrategien](#).

## Tagging-Einschränkungen für Amazon Keyspaces

Jedes Tag besteht aus einem Schlüssel und einem Wert, die Sie beide selbst definieren können. Beachten Sie die folgenden Einschränkungen:

- Jeder Amazon Keyspaces Wenn Sie versuchen, ein vorhandenes Tag (derselbe Schlüssel) hinzuzufügen, wird der vorhandene Tag-Wert auf den neuen Wert aktualisiert.
- Auf einen Schlüsselraum angewendete Tags gelten nicht automatisch für Tabellen innerhalb dieses Schlüsselraums. Um dasselbe Tag auf einen Schlüsselraum und alle seine Tabellen anzuwenden, muss jede Ressource einzeln gekennzeichnet werden.
- Wenn Sie einen Schlüsselraum oder eine Tabelle mit mehreren Regionen erstellen, werden alle Tags, die Sie während des Erstellungsprozesses definieren, automatisch auf alle Schlüsselräume und Tabellen in allen Regionen angewendet. Wenn Sie bestehende Tags mit `ALTER KEYSPACE` oder `ALTER TABLE` ändern, wird die Aktualisierung nur auf den Schlüsselraum oder die Tabelle in der Region angewendet, in der Sie die Änderung vornehmen.
- Ein Wert fungiert als Deskriptor in einer Tag-Kategorie (Schlüssel). In Amazon Keyspaces
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Die maximale Schlüssellänge beträgt 128 Unicode-Zeichen.
- Die maximale Wertlänge beträgt 256 Unicode-Zeichen.
- Namen dürfen Buchstaben, Leerzeichen und Zahlen sowie die folgenden Sonderzeichen enthalten:  
+ - = . \_ : /
- Die maximale Anzahl an Tags pro Ressource beträgt 50.
- AWS-Namen und Werte der von generierten Tags wird automatisch das `aws:` Präfix zugewiesen, das Sie nicht zuweisen können. AWS-zugewiesene Tag-Namen werden nicht auf das Tag-Limit von 50 angerechnet. Von Benutzern zugewiesene Tag-Namen haben das Präfix `user:` im Kostenzuordnungsbericht.
- Sie können die Anwendung eines Tags nicht rückdatieren.

## von Operationen für Amazon Keyspaces

Mithilfe der Amazon Keyspaces-Konsole (für Apache Cassandra), der AWS CLI oder der Cassandra Query Language (CQL) können Sie Tags für Keyspaces und Tabellen hinzufügen, auflisten, bearbeiten oder löschen. Anschließend können Sie diese benutzerdefinierten Tags aktivieren, damit sie in der AWS Billing and Cost Management-Konsole zur Nachverfolgung der Kostenzuordnung

erscheinen. Weitere Informationen finden Sie unter [Kostenzuordnungsberichte für Amazon Keyspaces](#).

Für die Massенbearbeitung können Sie auch den Tag Editor auf der Konsole verwenden. Weitere Informationen finden Sie unter [Arbeiten mit Tag Editor](#) im AWS Resource Groups-Benutzerhandbuch.

## Themen

- [Hinzufügen von über die Konsole zu neuen oder vorhandenen Schlüsselräumen und Tabellen](#)
- [Hinzufügen von über die AWS CLI zu neuen oder vorhandenen Schlüsselräumen und Tabellen](#)
- [Hinzufügen von von Spaces zu neuen oder vorhandenen Schlüsselräumen und Tabellen mit CQL](#)

## Hinzufügen von über die Konsole zu neuen oder vorhandenen Schlüsselräumen und Tabellen

Sie können die Amazon-Keyspaces-Konsole verwenden, um neuen Schlüsselräumen und Tabellen Tags hinzuzufügen, wenn Sie sie erstellen. Sie können auch Tags für vorhandene Tabellen hinzufügen, bearbeiten oder löschen.

Um Schlüsselräume zu markieren, wenn sie erstellt werden (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Keyspaces und dann Create von keyspace aus.
3. Geben Sie auf der Seite Keyspace erstellen einen Namen für den Keyspace ein. Geben Sie einen Schlüssel und den Wert für den Tag ein und wählen Sie dann Add new Tag (Neuen Tag hinzufügen) aus.
4. Wählen Sie Create keyspace.

Um Tabellen zu taggen, wenn sie erstellt werden (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tables (Tabellen) und anschließend Create table (Tabelle erstellen) aus.
3. Wählen Sie auf der Seite Tabelle erstellen im Abschnitt Tabellendetails ein Schlüsselfeld aus und geben Sie einen Namen für die Tabelle ein.

4. Erstellen Sie im Abschnitt Schema das Schema für Ihre Tabelle.
5. Wählen Sie im Abschnitt Tabelleneinstellungen die Option Einstellungen anpassen aus.
6. Fahren Sie mit dem Abschnitt Tabellentags — optional fort und wählen Sie Neues Tag hinzufügen, um neue Tags zu erstellen.
7. Wählen Sie Create table (Tabelle erstellen) aus.

#### Markieren vorhandener Ressourcen (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Keyspaces oder Tables aus.
3. Wählen Sie ein Schlüsselraum oder eine Tabelle in der Liste aus. Wählen Sie dann Tags verwalten aus, um Ihre Tags hinzuzufügen, zu bearbeiten oder zu löschen.

Informationen zur Tag-Struktur finden Sie unter [Tagging-Einschränkungen für Amazon Keyspaces](#).

## Hinzufügen von über die AWS CLI zu neuen oder vorhandenen Schlüsselräumen und Tabellen

Die Beispiele in diesem Abschnitt zeigen, wie Sie mithilfe der AWS CLI Tags angeben, wenn Sie Schlüsselräume und Tabellen erstellen, wie Sie Tags zu vorhandenen Ressourcen hinzufügen oder aus vorhandenen Ressourcen entfernen und wie Sie Tags auflisten.

Im folgenden Beispiel wird gezeigt, wie eine neue Tabelle mit Tags erstellt wird. Der Befehl erstellt eine Tabelle MyTable in einem bereits vorhandenen Keyspace MyKeySpace. Beachten Sie, dass der Befehl zur besseren Lesbarkeit in verschiedene Zeilen aufgeteilt wurde.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable'
 --schema-definition 'allColumns=[{name=id,type=int},{name=name,type=text},
{name=date,type=timestamp}],partitionKeys=[{name=id}]'
 --tags 'key=key1,value=val1' 'key=key2,value=val2'
```

Im folgenden Beispiel wird gezeigt, wie Sie einer vorhandenen Tabelle neue Tags hinzufügen.

```
aws keyspaces tag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111222333444:/
keyspace/myKeyspace/table/myTable' --tags 'key=key3,value=val3' 'key=key4,value=val4'
```

Im nächsten Beispiel wird gezeigt, wie Sie die Tags der angegebenen Ressource auflisten.

```
aws keyspaces list-tags-for-resource --resource-arn 'arn:aws:cassandra:us-east-1:111222333444:/keyspace/myKeyspace/table/myTable'
```

Die Ausgabe des letzten Befehls sieht so aus.

```
{
 "tags": [
 {
 "key": "key1",
 "value": "val1"
 },
 {
 "key": "key2",
 "value": "val2"
 },
 {
 "key": "key3",
 "value": "val3"
 },
 {
 "key": "key4",
 "value": "val4"
 }
]
}
```

## Hinzufügen von von Spaces zu neuen oder vorhandenen Schlüsselräumen und Tabellen mit CQL

Die folgenden Beispiele zeigen, wie Sie mithilfe der Tags angeben, wenn Sie Schlüsselräume und Tabellen erstellen, wie Sie vorhandene Ressourcen mit Tags versehen, und wie Sie Tags lesen.

Im folgenden Beispiel wird ein neuer Schlüsselraum mit Tags erstellt.

```
CREATE KEYSPACE mykeyspace WITH TAGS = {'key1':'val1', 'key2':'val2'} ;
```

Im folgenden Beispiel wird eine neue Tabelle mit Tags erstellt.



```
CREATE TABLE mytable(...) WITH TAGS = {'key1':'val1', 'key2':'val2'};
```

Um Ressourcen in einer Anweisung mit anderen Befehlen zu kennzeichnen.

```
CREATE KEYSPACE mykeyspace WITH REPLICATION = {'class': 'Simple Strategy'} AND TAGS
= {'key1':'val1', 'key2':'val2'};
```

Im folgenden Beispiel wird gezeigt, wie Sie Tags für vorhandene Schlüsselräume und Tabellen hinzufügen oder entfernen.

```
ALTER KEYSPACE mykeyspace ADD TAGS {'key1':'val1', 'key2':'val2'};
```

```
ALTER TABLE mytable DROP TAGS {'key1':'val1', 'key2':'val2'};
```

Verwenden Sie die folgende CQL-Anweisung, um die an eine Ressource angehängten Tags zu lesen.

```
SELECT * FROM system_schema_mcs.tags WHERE valid_where_clause;
```

Die WHERE Klausel ist erforderlich und muss eines der folgenden Formate aufweisen:

- `keyspace_name = 'mykeyspace' AND resource_type = 'keyspace'`
- `keyspace_name = 'mykeyspace' AND resource_name = 'mytable'`
- `resource_id = arn`

Beispiele:

Die folgende Abfrage zeigt, ob ein Keyspace über Tags verfügt.

```
SELECT * FROM system_schema_mcs.tags WHERE keyspace_name = 'mykeyspace' AND
resource_type = 'keyspace';
```

Die Ausgabe der Abfrage sieht wie folgt aus.

```

resource_id | keyspace_name |
resource_name | resource_type | tags
-----+-----
arn:aws:cassandra:us-east-1:123456789:/keyspace/mykeyspace/ | mykeyspace |
mykeyspace | keyspace | {'key1': 'val1', 'key2': 'val2'}

```

Die folgende Abfrage zeigt die Tags für eine Tabelle.

```

SELECT * FROM system_schema_mcs.tags WHERE keyspace_name = 'mykeyspace' AND
resource_name = 'mytable';

```

Die Ausgabe der Abfrage sieht wie folgt aus.

```

resource_id |
keyspace_name | resource_name | resource_type | tags
-----+-----+-----+-----
arn:aws:cassandra:us-east-1:123456789:/keyspace/mykeyspace/table/mytable |
mykeyspace | mytable | table | {'key1': 'val1', 'key2': 'val2'}

```

## Kostenzuordnungsberichte für Amazon Keyspaces

AWS verwendet Tags, um Ressourcenkosten in Ihrem Kostenzuordnungsbericht zu organisieren. AWS bietet zwei Arten von Tags für die Kostenzuordnung:

- Ein AWS-generierter-Tag. AWS definiert, erstellt und wendet dieses Tag für Sie an.
- Benutzerdefinierte Tags. Sie definieren und erstellen diese Tags und wenden sie an.

Sie müssen beide Arten von Tags separat aktivieren, bevor sie in Cost Explorer oder einem Kostenzuordnungsbericht angezeigt werden können.


So aktivieren Sie von AWS generierte Tags:

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die Fakturierungs- und Kostenverwaltungskonsole unter <https://console.aws.amazon.com/billing/home#/>.
2. Wählen Sie im Navigationsbereich die Option Cost Allocation Tags (Kostenzuordnungs-Tags) aus.
3. Klicken Sie unter AWS-Generated Cost Allocation Tags (generierte Kostenzuordnungs-Tags) und wählen Sie die Option Enable (Aktivieren).

So aktivieren Sie benutzerdefinierte Tags:

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die Fakturierungs- und Kostenverwaltungskonsole unter <https://console.aws.amazon.com/billing/home#/>.
2. Wählen Sie im Navigationsbereich die Option Cost Allocation Tags (Kostenzuordnungs-Tags) aus.
3. Wählen Sie unter User-Defined Cost Allocation Tags (Benutzerdefinierte Kostenzuordnungs-Tags) die Option Activate (Aktivieren) aus.

Nach dem Erstellen und Aktivieren der Tags generiert AWS einen Kostenzuordnungsbericht, in dem Ihre Nutzungsdaten und Kosten nach Ihren aktiven Tags gruppiert sind. Der Kostenzuordnungsbericht enthält alle Ihre AWS-Kosten für jeden Abrechnungszeitraum. Der Bericht enthält getaggte und nicht getaggte Ressourcen, sodass Sie die Gebühren für Ressourcen übersichtlich verwalten können.

 Note

Derzeit werden alle Daten, die aus Amazon Keyspaces weitergeleitet werden, im Kostenzuordnungsbericht nicht nach Tags aufgeschlüsselt.

Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#).

# Bewährte Methoden für Design und Architektur mit Amazon Keyspaces

In diesem Abschnitt finden Sie schnell Empfehlungen zur Maximierung der Leistung und Minimierung der Durchsatzkosten bei der Arbeit mit Amazon Keyspaces.

## Inhalt

- [NoSQL-Design für Amazon Keyspaces](#)
  - [Unterschiede zwischen einem relationalen Datendesign und NoSQL](#)
  - [Zwei Schlüsselkonzepte für das NoSQL-Design](#)
  - [Ansatz für ein NoSQL-Design](#)
- [Client-Treiberverbindungen zu Amazon Keyspaces \(für Apache Cassandra\)](#)
  - [So funktionieren Verbindungen in Amazon Keyspaces](#)
  - [So konfigurieren Sie Verbindungen in Amazon Keyspaces](#)
  - [So konfigurieren Sie Verbindungen über VPC-Endpunkte in Amazon Keyspaces](#)
  - [So überwachen Sie Verbindungen in Amazon Keyspaces](#)
  - [So behandeln Sie Verbindungsfehler in Amazon Keyspaces](#)
- [Datenmodellierung in Amazon Keyspaces \(für Apache Cassandra\)](#)
  - [So verwenden Sie Partitionsschlüssel effektiv in Amazon Keyspaces](#)
    - [Verwenden von Write-Sharding zur gleichmäßigen Verteilung von Workloads in Amazon Keyspaces](#)
      - [Sharding mit zusammengesetzten Partitionsschlüsseln und Zufallswerten](#)
      - [Sharding mit zusammengesetzten Partitionsschlüsseln und berechneten Werten](#)
- [Optimierung der Kosten von Amazon Keyspaces-Tabellen](#)
  - [Auswerten Ihrer Kosten auf Tabellenebene](#)
    - [So zeigen Sie die Kosten einer einzelnen Amazon Keyspaces-Tabelle an](#)
    - [Standardansicht von Cost Explorer](#)
    - [So verwenden Sie Tabellen-Tags in Cost Explorer](#)
  - [Evaluieren Sie den Kapazitätsmodus Ihrer Tabelle](#)
    - [Verfügbare Tabellenkapazitätsmodi](#)
    - [Fälle, in denen der On-Demand-Modus ausgewählt werden sollte](#)

- [Fälle, in denen der Modus bereitgestellter Kapazität ausgewählt werden sollte](#)
- [Zusätzliche Faktoren, die bei der Auswahl eines Tabellenkapazitätsmodus zu berücksichtigen sind](#)
- [Evaluieren Sie die Application Auto Scaling Scaling-Einstellungen Ihrer Tabelle](#)
  - [Grundlegendes zu Ihren Application Auto Scaling Scaling-Einstellungen](#)
  - [So ermitteln Sie Tabellen mit geringer Zielauslastung \(<= 50 %\)](#)
  - [So bewältigen Sie Workloads mit saisonalen Schwankungen](#)
  - [So bewältigen Sie stark schwankende Workloads mit unbekanntem Mustern](#)
  - [So bewältigen Sie Workloads mit verknüpften Anwendungen](#)
- [Identifizieren Sie Ihre ungenutzten Ressourcen](#)
  - [So ermitteln Sie nicht verwendete Ressourcen](#)
  - [Ermitteln von nicht verwendeten Tabellenressourcen](#)
  - [Bereinigen von nicht verwendeten Tabellenressourcen](#)
  - [Bereinigen ungenutzter point-in-time Wiederherstellungs-Backups \(PITR\)](#)
- [Auswerten Ihrer Tabellennutzungsmuster](#)
  - [Ausführen von weniger strikt konsistenten Lesevorgängen](#)
  - [Aktivieren von Time to Live \(TTL\)](#)
- [Auswerten Ihrer bereitgestellten Kapazität, um eine angemessene Bereitstellung zu erzielen](#)
  - [So rufen Sie Verbrauchsmetriken aus Ihren Amazon Keyspaces-Tabellen ab](#)
  - [So identifizieren Sie Amazon Keyspaces-Tabellen mit unzureichender Bereitstellung](#)
  - [So identifizieren Sie übermäßig bereitgestellte Amazon Keyspaces-Tabellen](#)

## NoSQL-Design für Amazon Keyspaces

NoSQL-Datenbanksysteme wie Amazon Keyspaces verwenden alternative Modelle für das Datenmanagement, z. B. Schlüssel-Wert-Paare oder die Speicherung von Dokumenten. Wenn Sie von einem relationalen Datenbankmanagementsystem zu einem NoSQL-Datenbanksystem wie Amazon Keyspaces wechseln, ist es wichtig, die wichtigsten Unterschiede und spezifischen Entwurfsansätze zu verstehen.

### Themen

- [Zwei Schlüsselkonzepte für das NoSQL-Design](#)
- [Ansatz für ein NoSQL-Design](#)

## Unterschiede zwischen einem relationalen Datendesign und NoSQL

Relationale Datenbankmanagementsysteme (RDBMS) und nNoSQL-Datenbanken besitzen unterschiedliche Vor- und Nachteile:

- In RDBMS können Daten flexibel abgerufen werden, Abfragen sind jedoch vergleichsweise kostspielig und können bei einer großen Zahl von Zugriffen nicht gut skaliert werden (siehe [the section called “Datenmodellierung”](#)).
- In einer NoSQL-Datenbank wie Amazon Keyspaces können Daten auf eine begrenzte Anzahl von Arten effizient abgefragt werden. Andernfalls können Abfragen teuer und langsam sein.

Diese Unterschiede führen dazu, dass das Datenbankdesign der beiden Systeme verschieden ist:

- In RDBMS entwerfen Sie das Design im Hinblick auf Flexibilität, ohne sich um Implementierungsdetails oder Leistung zu kümmern. Die Optimierung von Abfragen wirkt sich im Allgemeinen nicht auf das Schemadesign aus, eine Standardisierung ist jedoch wichtig.
- In Amazon Keyspaces entwerfen Sie Ihr Schema speziell, um die häufigsten und wichtigsten Abfragen so schnell und kostengünstig wie möglich zu gestalten. Ihre Datenstrukturen sind an die spezifischen Anforderungen Ihrer geschäftlichen Anwendungsfälle angepasst.

## Zwei Schlüsselkonzepte für das NoSQL-Design

Das NoSQL-Design erfordert einen anderen Ansatz als das RDBMS-Design. Sie können für ein RDBMS ein standardisiertes Datenmodell entwickeln, ohne sich Gedanken über Zugriffsmuster machen zu müssen. Anschließend können Sie es erweitern, wenn neue Fragen und Abfrageanforderungen entstehen. Sie können jeden einzelnen Typ von Daten in einer eigenen Tabelle organisieren.

Wie sich das NoSQL-Design unterscheidet

- Im Gegensatz dazu sollten Sie nicht mit dem Entwerfen Ihres Schemas für Amazon Keyspaces beginnen, bis Sie wissen, welche Fragen es beantworten muss. Es ist äußerst wichtig, die geschäftlichen Probleme und Anwendungsfälle vor der Entwicklung des Schemas zu kennen.

- Sie sollten so wenige Tabellen wie möglich in einer Amazon Keyspaces-Anwendung verwalten. Weniger Tabellen sorgen für mehr Skalierbarkeit, erfordern weniger Rechteverwaltung und reduzieren den Overhead für Ihre Amazon Keyspaces-Anwendung. Dies kann auch dazu beitragen, die Backup-Kosten insgesamt niedrig zu halten.

## Ansatz für ein NoSQL-Design

Der erste Schritt beim Entwerfen Ihrer Amazon Keyspaces-Anwendung besteht darin, die spezifischen Abfragemuster zu identifizieren, die das System erfüllen muss.

Insbesondere ist es wichtig, drei Basiseigenschaften der Zugriffsmuster Ihrer Anwendung zu kennen, bevor Sie mit dem Design beginnen:

- **Datengröße:** Wenn Sie wissen, wie viele Daten gleichzeitig gespeichert und angefordert werden, können Sie ermitteln, wie die Daten am effektivsten partitioniert werden können.
- **Datenform:** Statt Daten neu zu gestalten, wenn eine Abfrage verarbeitet wird (wie im Fall von RDBMS-Systemen), organisieren NoSQL-Datenbanken Daten so, dass ihre Form in der Datenbank dem entspricht, was abgefragt werden wird. Dies ist ein entscheidender Faktor, um Geschwindigkeit und Skalierbarkeit zu verbessern.
- **Datengeschwindigkeit:** Amazon Keyspaces skaliert, indem es die Anzahl der physischen Partitionen erhöht, die für die Verarbeitung von Abfragen verfügbar sind, und indem Daten effizient auf diese Partitionen verteilt werden. Wenn Sie im Voraus wissen, wie die Spitzenabfragelasten aussehen könnten, hilft Ihnen dies möglicherweise, die Daten so zu partitionieren, dass die I/O-Kapazität optimal genutzt wird.

Nach der Identifizierung spezifischer Abfrageanforderungen können Sie die Daten nach allgemeinen Grundsätzen organisieren, denen die Leistung unterliegt:

- **Speichern Sie verwandte Daten zusammen.** Studien zur Optimierung von Routing-Tabellen, die vor 20 Jahren durchgeführt wurden, hatten zum Ergebnis, dass der "Referenzspeicherort" den wichtigsten Faktor für die Verkürzung der Reaktionszeit darstellt: Daten sollten daher zentral gespeichert werden. Das gilt auch heute für NoSQL-Systeme: Wenn verwandte Daten nahe beieinander gespeichert werden, hat dies große Auswirkungen auf Kosten und Leistung. Statt verwandte Datenelemente über mehrere Tabellen zu verteilen, sollten Sie verwandte Elemente in Ihrem NoSQL-System so nahe beieinander wie möglich speichern.

In der Regel sollten Sie in einer Amazon Keyspaces-Anwendung so wenige Tabellen wie möglich verwalten.

Ausnahmen hiervon sind Fälle, in denen große Mengen von Zeitreihendaten oder Datensätze mit sehr unterschiedlichen Zugriffsmustern vorhanden sind. Eine einzelne Tabelle mit umgekehrten Indizes kann in der Regel einfache Abfragen unterstützen, um die komplexen hierarchischen Datenstrukturen zu erstellen und abzurufen, die Ihre Anwendung benötigt.

- Verwenden Sie eine Sortierreihenfolge. Verwandte Elemente können zusammen gruppiert und effizient abgefragt werden, wenn ihr Schlüsseldesign dazu führt, dass sie gemeinsam sortiert werden. Dies stellt eine wichtige Strategie für das NoSQL-Design dar.
- Verteilen Sie Abfragen. Es ist außerdem wichtig, zu vermeiden, eine große Zahl von Abfragen auf einen einzigen Teil der Datenbank auszurichten, da dies die E/A-Kapazität überschreiten könnte. Sie sollten stattdessen Datenschlüssel entwerfen, um die Zugriffe so gleichmäßig wie möglich über die Partitionen zu verteilen und Konzentrationspunkte zu vermeiden.

Diese allgemeinen Prinzipien lassen sich in einigen gängigen Entwurfsmustern umsetzen, die Sie verwenden können, um Daten in Amazon Keyspaces effizient zu modellieren.

## Client-Treiberverbindungen zu Amazon Keyspaces (für Apache Cassandra)

Für die Kommunikation mit Amazon Keyspaces können Sie einen der vorhandenen Apache Cassandra-Client-Treiber Ihrer Wahl verwenden. Da es sich bei Amazon Keyspaces um einen serverlosen Service handelt, empfehlen wir Ihnen, die Verbindungskonfiguration Ihres Client-Treibers für die Durchsatzanforderungen Ihrer Anwendung zu optimieren. In diesem Thema werden bewährte Methoden vorgestellt, darunter die Berechnung, wie viele Verbindungen Ihre Anwendung benötigt, sowie die Überwachung und Fehlerbehandlung von Verbindungen.

### Themen

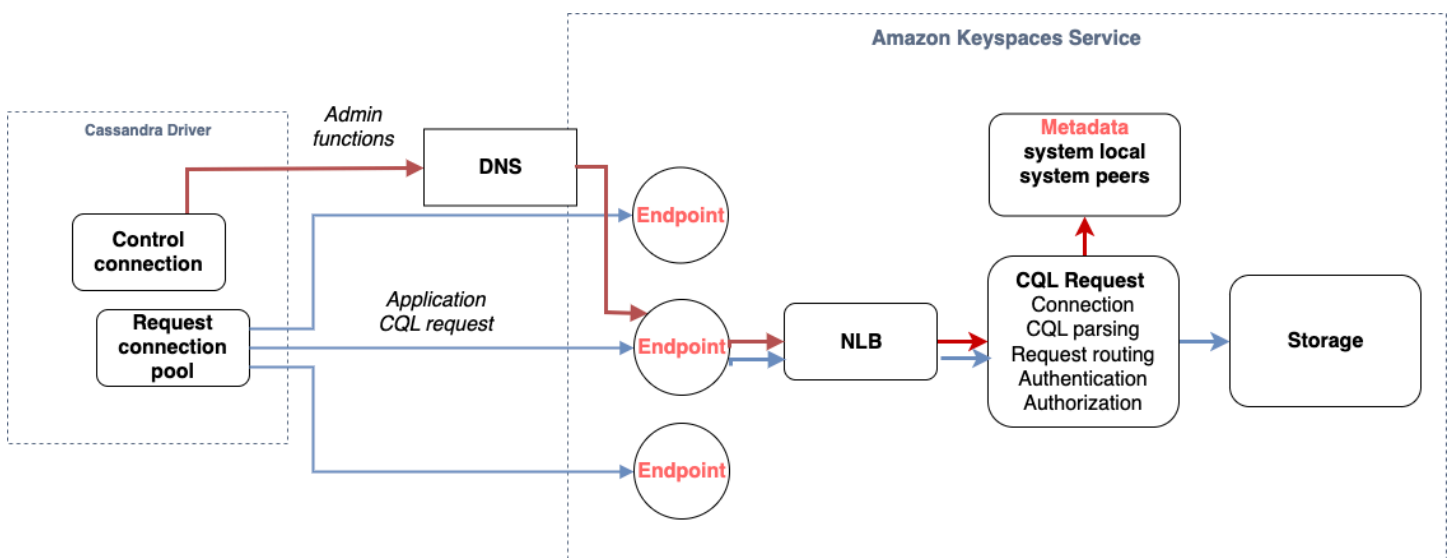
- [So funktionieren Verbindungen in Amazon Keyspaces](#)
- [So konfigurieren Sie Verbindungen in Amazon Keyspaces](#)
- [So konfigurieren Sie Verbindungen über VPC-Endpunkte in Amazon Keyspaces](#)
- [So überwachen Sie Verbindungen in Amazon Keyspaces](#)
- [So behandeln Sie Verbindungsfehler in Amazon Keyspaces](#)



## So funktionieren Verbindungen in Amazon Keyspaces

Dieser Abschnitt gibt einen Überblick darüber, wie Client-Treiberverbindungen in Amazon Keyspaces funktionieren. Da eine Fehlkonfiguration des Cassandra-Clienttreibers zu `PerConnectionRequestExceeded` Ereignissen in Amazon Keyspaces führen kann, ist die Konfiguration der richtigen Anzahl von Verbindungen in der Client-Treiberkonfiguration erforderlich, um diese und ähnliche Verbindungsfehler zu vermeiden.

Bei der Verbindung mit Amazon Keyspaces benötigt der Treiber einen Seed-Endpoint, um eine erste Verbindung herzustellen. Amazon Keyspaces verwendet DNS, um die erste Verbindung zu einem der vielen verfügbaren Endpunkte weiterzuleiten. Die Endpunkte sind mit Netzwerk-Loadbalancern verbunden, die wiederum eine Verbindung zu einem der Request-Handler in der Flotte herstellen. Nachdem die erste Verbindung hergestellt wurde, sammelt der Client-Treiber Informationen über alle verfügbaren Endpunkte aus der Tabelle `system.peers`. Mit diesen Informationen kann der Client-Treiber zusätzliche Verbindungen zu den aufgelisteten Endpunkten herstellen. Die Anzahl der Verbindungen, die der Clienttreiber herstellen kann, ist durch die Anzahl der lokalen Verbindungen begrenzt, die in den Einstellungen des Clienttreibers angegeben ist. Standardmäßig stellen die meisten Clienttreiber eine Verbindung pro Endpunkt her und richten einen Verbindungspool zu Cassandra ein und verteilen Abfragen über diesen Verbindungspool. Obwohl mehrere Verbindungen zu demselben Endpunkt hergestellt werden können, können sie hinter dem Network Load Balancer mit vielen verschiedenen Request-Handlern verbunden sein. Wenn Sie eine Verbindung über den öffentlichen Endpunkt herstellen, führt der Aufbau einer Verbindung zu jedem der neun in der `system.peers` Tabelle aufgeführten Endpunkte zu neun Verbindungen zu unterschiedlichen Request-Handlern.



## So konfigurieren Sie Verbindungen in Amazon Keyspaces

Amazon Keyspaces unterstützt bis zu 3.000 CQL-Abfragen pro TCP-Verbindung pro Sekunde. Da die Anzahl der Verbindungen, die ein Treiber herstellen kann, unbegrenzt ist, empfehlen wir, nur 500 CQL-Anfragen pro Sekunde und Verbindung anzuvisieren, um Overhead, Traffic-Bursts und einen besseren Lastenausgleich zu berücksichtigen. Gehen Sie wie folgt vor, um sicherzustellen, dass die Treiberverbindung korrekt für die Anforderungen Ihrer Anwendung konfiguriert ist.

Erhöhen Sie die Anzahl der Verbindungen pro IP-Adresse, die Ihr Treiber in seinem Verbindungspool verwaltet.

- Die meisten Cassandra-Treiber richten einen Verbindungspool zu Cassandra ein und verteilen Abfragen über diesen Verbindungspool. Das Standardverhalten der meisten Treiber besteht darin, eine einzige Verbindung zu jedem Endpunkt herzustellen. Amazon Keyspaces stellt Treibern neun Peer-IP-Adressen zur Verfügung. Basierend auf dem Standardverhalten der meisten Treiber führt dies zu 9 Verbindungen. Amazon Keyspaces unterstützt bis zu 3.000 CQL-Abfragen pro TCP-Verbindung pro Sekunde. Daher beträgt der maximale CQL-Abfragedurchsatz eines Treibers mit den Standardeinstellungen 27.000 CQL-Abfragen pro Sekunde. Wenn Sie die Standardeinstellungen des Treibers verwenden, muss eine einzelne Verbindung möglicherweise mehr als den maximalen CQL-Abfragedurchsatz von 3.000 CQL-Abfragen pro Sekunde verarbeiten. Dies könnte zu Ereignissen führen. `PerConnectionRequestExceeded`
- Um `PerConnectionRequestExceeded` Ereignisse zu vermeiden, müssen Sie den Treiber so konfigurieren, dass zusätzliche Verbindungen pro Endpunkt hergestellt werden, um den Durchsatz zu verteilen.
- Als bewährte Methode in Amazon Keyspaces wird davon ausgegangen, dass jede Verbindung 500 CQL-Abfragen pro Sekunde unterstützen kann.
- Das bedeutet, dass Sie für eine Produktionsanwendung, die schätzungsweise 27.000 CQL-Abfragen pro Sekunde, verteilt auf die neun verfügbaren Endpunkte, unterstützen muss, sechs Verbindungen pro Endpunkt konfigurieren müssen. Dadurch wird sichergestellt, dass jede Verbindung nicht mehr als 500 Anfragen pro Sekunde verarbeitet.

Berechnen Sie die Anzahl der Verbindungen pro IP-Adresse, die Sie für Ihren Treiber konfigurieren müssen, basierend auf den Anforderungen Ihrer Anwendung.

Betrachten Sie das folgende Beispiel, um die Anzahl der Verbindungen zu ermitteln, die Sie pro Endpunkt für Ihre Anwendung konfigurieren müssen. Sie haben eine Anwendung, die 20.000 CQL-Abfragen pro Sekunde unterstützen muss, die aus 10.000 INSERTSELECT, 5.000 und 5.000

DELETE Vorgängen bestehen. Die Java-Anwendung wird auf drei Instances auf Amazon Elastic Container Service (Amazon ECS) ausgeführt, wobei jede Instance eine einzelne Sitzung mit Amazon Keyspaces einrichtet. Die Berechnung, mit der Sie abschätzen können, wie viele Verbindungen Sie für Ihren Treiber konfigurieren müssen, verwendet die folgende Eingabe.

1. Die Anzahl der Anfragen pro Sekunde, die Ihre Anwendung unterstützen muss.
2. Die Anzahl der verfügbaren Instances, wobei eine Instanz abgezogen wird, um Wartungs- oder Ausfallzeiten zu berücksichtigen.
3. Die Anzahl der verfügbaren Endpunkte. Wenn Sie eine Verbindung über öffentliche Endpunkte herstellen, stehen Ihnen neun Endpunkte zur Verfügung. Wenn Sie VPC-Endpunkte verwenden, stehen Ihnen je nach Region zwischen zwei und fünf Endpunkte zur Verfügung.
4. Verwenden Sie 500 CQL-Abfragen pro Sekunde und Verbindung als bewährte Methode für Amazon Keyspaces.
5. Runden Sie das Ergebnis zusammen.

In diesem Beispiel sieht die Formel so aus.

```
20,000 CQL queries / (3 instances - 1 failure) / 9 public endpoints / 500 CQL queries per second = ROUND(2.22) = 3
```

Basierend auf dieser Berechnung müssen Sie in der Treiberkonfiguration drei lokale Verbindungen pro Endpunkt angeben. Für Remoteverbindungen konfigurieren Sie nur eine Verbindung pro Endpunkt.

## So konfigurieren Sie Verbindungen über VPC-Endpunkte in Amazon Keyspaces

Wenn Sie eine Verbindung über private VPC-Endpunkte herstellen, stehen Ihnen höchstwahrscheinlich 3 Endpunkte zur Verfügung. Die Anzahl der VPC-Endpunkte kann je nach Region unterschiedlich sein, basierend auf der Anzahl der Availability Zones und der Anzahl der Subnetze in der zugewiesenen VPC. Die Region USA Ost (Nord-Virginia) hat fünf Availability Zones und Sie können bis zu fünf Amazon Keyspaces-Endpunkte haben. Die Region USA West (Nordkalifornien) hat zwei Availability Zones und Sie können bis zu zwei Amazon Keyspaces-Endpunkte haben. Die Anzahl der Endpunkte hat keinen Einfluss auf die Skalierung, erhöht jedoch die Anzahl der Verbindungen, die Sie in der Treiberkonfiguration einrichten müssen. Betrachten Sie das folgende Beispiel. Ihre Anwendung muss 20.000 CQL-Abfragen unterstützen und wird

auf drei Instances auf Amazon ECS ausgeführt, wobei jede Instance eine einzelne Sitzung zu Amazon Keyspaces einrichtet. Der einzige Unterschied besteht darin, wie viele Endpunkte in den verschiedenen verfügbar sind. AWS-Regionen

In der Region USA Ost (Nord-Virginia) sind Verbindungen erforderlich:

```
20,000 CQL queries / (3 instances - 1 failure) / 5 private VPC endpoints / 500 CQL
queries per second = 4 local connections
```

In der Region USA West (Nordkalifornien) sind Verbindungen erforderlich:

```
20,000 CQL queries / (3 instances - 1 failure) / 2 private VPC endpoints / 500 CQL
queries per second = 10 local connections
```

### Important

Bei der Verwendung von privaten VPC-Endpunkten sind zusätzliche Berechtigungen erforderlich, damit Amazon Keyspaces die verfügbaren VPC-Endpunkte dynamisch ermitteln und die Tabelle auffüllen kann. `system.peers` Weitere Informationen finden Sie unter [the section called “Auffüllen von `system.peers` Tabelleneinträgen mit VPC-Endpunktinformationen der Schnittstelle”](#).

Wenn Sie über einen privaten VPC-Endpunkt mit einem anderen auf Amazon Keyspaces zugreifen, ist es wahrscheinlich AWS-Konto, dass Sie nur einen einzigen Amazon Keyspaces-Endpunkt sehen. Auch dies hat keinen Einfluss auf den Umfang des möglichen Durchsatzes zu Amazon Keyspaces, aber es kann erforderlich sein, dass Sie die Anzahl der Verbindungen in Ihrer Treiberkonfiguration erhöhen. Dieses Beispiel zeigt dieselbe Berechnung für einen einzelnen verfügbaren Endpunkt.

```
20,000 CQL queries / (3 instances - 1 failure) / 1 private VPC endpoints / 500 CQL
queries per second = 20 local connections
```

Weitere Informationen zum kontoübergreifenden Zugriff auf Amazon Keyspaces über eine gemeinsam genutzte VPC finden Sie unter [the section called “Kontoübergreifende Zugriff”](#)

## So überwachen Sie Verbindungen in Amazon Keyspaces

Um die Anzahl der Endpunkte zu ermitteln, mit denen Ihre Anwendung verbunden ist, können Sie die Anzahl der erkannten Peers in der `system.peers` Tabelle protokollieren. Das folgende Beispiel ist

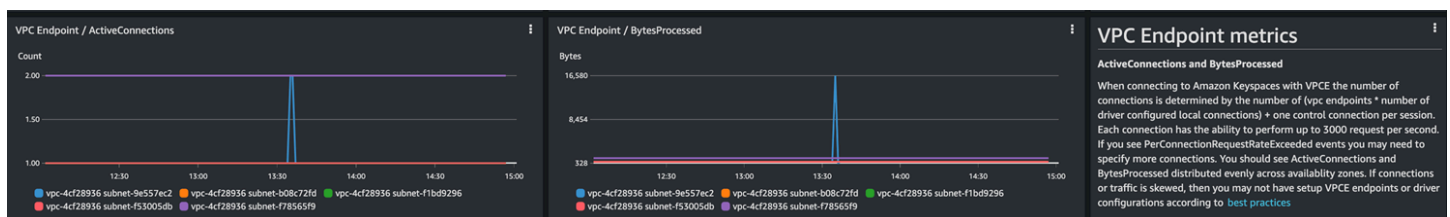
ein Beispiel für Java-Code, der die Anzahl der Peers ausgibt, nachdem die Verbindung hergestellt wurde.

```
ResultSet result = session.execute(new SimpleStatement("SELECT * FROM system.peers"));
logger.info("number of Amazon Keyspaces endpoints:" + result.all().stream().count());
```

### Note

Die CQL-Konsole oder AWS -Konsole werden nicht in einer VPC bereitgestellt und verwenden daher den öffentlichen Endpunkt. Wenn die `system.peers` Abfrage von Anwendungen aus ausgeführt wird, die sich außerhalb des VPCE befinden, führt dies daher häufig zu 9 Peers. Es kann auch hilfreich sein, die IP-Adressen der einzelnen Peers auszudrucken.

Sie können auch die Anzahl der Peers beobachten, wenn Sie einen VPC-Endpunkt verwenden, indem Sie CloudWatch VPCE-Amazon-Metriken einrichten. In können Sie die Anzahl der Verbindungen sehen CloudWatch, die zum VPC-Endpunkt hergestellt wurden. Die Cassandra-Treiber stellen für jeden Endpunkt eine Verbindung her, um CQL-Abfragen zu senden, und eine Kontrollverbindung, um Systemtabelleninformationen zu sammeln. Das Bild unten zeigt die CloudWatch VPC-Endpunktmetriken nach dem Herstellen einer Verbindung zu Amazon Keyspaces mit einer in den Treibereinstellungen konfigurierten Verbindung. Die Metrik zeigt sechs aktive Verbindungen, bestehend aus einer Kontrollverbindung und fünf Verbindungen (eine pro Endpunkt in allen Availability Zones).



Um mit der Überwachung der Anzahl der Verbindungen mithilfe eines CloudWatch Diagramms zu beginnen, können Sie diese AWS CloudFormation Vorlage bereitstellen, die GitHub im [Amazon Keyspaces-Vorlagen-Repository](#) verfügbar ist.

## So behandeln Sie Verbindungsfehler in Amazon Keyspaces

Wenn das Kontingent von 3.000 Anfragen pro Verbindung überschritten wird, gibt Amazon Keyspaces ein `PerConnectionRequestExceeded` Ereignis zurück und der Cassandra-Treiber erhält eine `WriteTimeout` `ReadTimeout` OR-Ausnahme. Sie sollten diese Ausnahme mit exponentiellem Backoff in Ihrer Cassandra-Wiederholungsrichtlinie oder in Ihrer Anwendung erneut versuchen. Sie sollten einen exponentiellen Backoff angeben, um zu vermeiden, dass zusätzliche Anfragen gesendet werden.

Die standardmäßige Wiederholungsrichtlinie versucht, dies `try next host` im Abfrageplan zu tun. Da Amazon Keyspaces beim Herstellen einer Verbindung zum VPC-Endpunkt über ein bis drei verfügbare Endpunkte verfügen kann, werden `NoHostAvailableException` in Ihren Anwendungsprotokollen möglicherweise auch die `ReadTimeout` Ausnahmen zusätzlich zu den `WriteTimeout` und angezeigt. Sie können die von Amazon Keyspaces bereitgestellten Wiederholungsrichtlinien verwenden, die es auf demselben Endpunkt, aber über verschiedene Verbindungen hinweg wiederholen.

Beispiele für exponentielle Wiederholungsrichtlinien für Java finden Sie GitHub im [Amazon Keyspaces-Repository für Java-Codebeispiele](#). Weitere Sprachbeispiele finden Sie auf Github im [Amazon Keyspaces-Repository für Codebeispiele](#).

## Datenmodellierung in Amazon Keyspaces (für Apache Cassandra)

In diesem Thema werden Datenmodellierungskonzepte in Amazon Keyspaces (für Apache Cassandra) vorgestellt. In diesem Abschnitt finden Sie Empfehlungen für den Entwurf von Datenmodellen, die den Datenzugriffsmustern Ihrer Anwendung entsprechen. Die Implementierung von Best Practices für die Datenmodellierung verbessert die Leistung und minimiert die Durchsatzkosten bei der Arbeit mit Amazon Keyspaces.

Um Datenmodelle einfacher zu visualisieren und zu entwerfen, können Sie die [NoSQL Workbench](#) verwenden.

### Themen

- [So verwenden Sie Partitionsschlüssel effektiv in Amazon Keyspaces](#)

## So verwenden Sie Partitionsschlüssel effektiv in Amazon Keyspaces

Der Primärschlüssel, der jede Zeile in einer Amazon Keyspaces-Tabelle eindeutig identifiziert, kann aus einer oder mehreren Partitionsschlüsselspalten bestehen, die bestimmen, in welchen Partitionen die Daten gespeichert werden, und einer oder mehreren optionalen Clusterspalten, die definieren, wie Daten innerhalb einer Partition geclustert und sortiert werden.

Da der Partitionsschlüssel die Anzahl der Partitionen festlegt, in denen Ihre Daten gespeichert sind und wie die Daten auf diese Partitionen verteilt werden, kann die Art und Weise, wie Sie Ihren Partitionsschlüssel auswählen, erhebliche Auswirkungen auf die Leistung Ihrer Abfragen haben. Im Allgemeinen sollten Sie Ihre Anwendung so entwerfen, dass sie auf allen Partitionen auf der Festplatte einheitlich abläuft.

Die gleichmäßige Verteilung der Lese- und Schreibaktivitäten Ihrer Anwendung auf alle Partitionen trägt zur Minimierung der Durchsatzkosten bei. Dies gilt sowohl für bedarfsgesteuerte als auch für bereitgestellte Lese-/Schreibkapazitätsmodi. Wenn Sie beispielsweise den Modus mit bereitgestellter Kapazität verwenden, können Sie die Zugriffsmuster ermitteln, die Ihre Anwendung benötigt, und die Gesamtzahl der Lesekapazitätseinheiten (RCU) und Schreibkapazitätseinheiten (WCU) schätzen, die jede Tabelle benötigt. Amazon Keyspaces unterstützt Ihre Zugriffsmuster mithilfe des von Ihnen bereitgestellten Durchsatzes, sofern der Datenverkehr auf einer bestimmten Partition 3.000 RCUs und 1.000 WCUs nicht überschreitet.

Amazon Keyspaces bietet zusätzliche Flexibilität bei der Bereitstellung von Durchsätzen pro Partition, indem es Burst-Kapazität bereitstellt. Weitere Informationen finden Sie unter [the section called “Burst-Kapazität”](#)

### Themen

- [Verwenden von Write-Sharding zur gleichmäßigen Verteilung von Workloads in Amazon Keyspaces](#)

## Verwenden von Write-Sharding zur gleichmäßigen Verteilung von Workloads in Amazon Keyspaces

Eine Möglichkeit, Schreibvorgänge besser auf eine Partition in Amazon Keyspaces zu verteilen, besteht darin, den Speicherplatz zu erweitern. Dies kann auf verschiedene Arten geschehen. Sie können eine zusätzliche Partitionsschlüsselspalte hinzufügen, in die Sie Zufallszahlen schreiben, um die Zeilen auf die Partitionen zu verteilen. Oder Sie können eine Zahl verwenden, die basierend auf einer Abfrage berechnet wird.



## Sharding mit zusammengesetzten Partitionsschlüsseln und Zufallswerten

Eine Strategie, um Lasten gleichmäßiger auf eine Partition zu verteilen, besteht darin, eine zusätzliche Partitionsschlüsselspalte hinzuzufügen, in die Sie Zufallszahlen schreiben. Anschließend werden die Schreibvorgänge zufällig auf den größeren Raum verteilt.

Stellen Sie sich zum Beispiel die folgende Tabelle vor, in der ein einziger Partitionsschlüssel ein Datum darstellt.

```
CREATE TABLE IF NOT EXISTS tracker.blogs (
 publish_date date,
 title text,
 description int,
 PRIMARY KEY (publish_date));
```

Um diese Tabelle gleichmäßiger auf die Partitionen zu verteilen, könnten Sie eine zusätzliche Partitionsschlüsselspalte `shard` hinzufügen, in der Zufallszahlen gespeichert werden. Beispielsweise:

```
CREATE TABLE IF NOT EXISTS tracker.blogs (
 publish_date date,
 shard int,
 title text,
 description int,
 PRIMARY KEY ((publish_date, shard)));
```

Beim Einfügen von Daten können Sie eine Zufallszahl zwischen 1 und 200 für die `shard` Spalte wählen. Dies ergibt zusammengesetzte Partitionsschlüsselwerte wie `(2020-07-09, 1)`, `(2020-07-09, 2)`, und so weiter, bis `(2020-07-09, 200)`. Durch die Randomisierung des Partitionsschlüssels werden die Schreibvorgänge in die Tabelle jeden Tag gleichmäßig auf mehrere Partitionen verteilt. Dies führt zu einer besseren Parallelverarbeitung und einem höheren Gesamtdurchsatz.

Um jedoch alle Zeilen für einen bestimmten Tag zu lesen, müssten Sie die Zeilen nach allen Shards abfragen und dann die Ergebnisse zusammenführen. Beispielsweise würden Sie zuerst eine `SELECT` Anweisung für den Wert `(2020-07-09, 1)` des Partitionsschlüssels ausgeben. Geben Sie dann eine weitere `SELECT` Anweisung für `(2020-07-09, 2)` usw. aus `(2020-07-09, 200)`. Schließlich müsste Ihre Anwendung die Ergebnisse all dieser `SELECT` Anweisungen zusammenführen.



## Sharding mit zusammengesetzten Partitionsschlüsseln und berechneten Werten

Mit einer Randomisierungsstrategie kann der Schreibdurchsatz erheblich verbessert werden. Es ist jedoch schwierig, eine bestimmte Zeile zu lesen, da Sie nicht wissen, welcher Wert in die `shard` Spalte geschrieben wurde, als die Zeile geschrieben wurde. Um das Lesen einzelner Zeilen zu erleichtern, können Sie eine andere Strategie verwenden. Anstatt eine Zufallszahl zu verwenden, um die Zeilen auf die Partitionen zu verteilen, verwenden Sie eine Zahl, die Sie auf der Grundlage eines Objekts berechnen können, das Sie abfragen möchten.

Denken Sie an das vorherige Beispiel, in dem eine Tabelle das Datum des aktuellen Tages im Partitionsschlüssel verwendet. Nehmen wir nun an, dass jede Zeile über eine `title` Spalte verfügt, auf die zugegriffen werden kann, und dass Sie Zeilen in den meisten Fällen nicht nur nach Datum, sondern auch nach Titel suchen müssen. Bevor Ihre Anwendung die Zeile in die Tabelle schreibt, könnte sie anhand des Titels einen Hashwert berechnen und diesen zum Füllen der `shard` Spalte verwenden. Die Berechnung könnte eine Zahl zwischen 1 und 200 ergeben, die ähnlich wie bei der Zufallsstrategie relativ gleichmäßig verteilt ist.

Eine einfache Berechnung würde wahrscheinlich ausreichen, z. B. das Produkt der UTF-8-Codepunktwerte für die Zeichen im Titel, Modulo 200, + 1. Der Wert des zusammengesetzten Partitionsschlüssels wäre dann die Kombination aus Datum und Berechnungsergebnis.

Mit dieser Strategie werden die Schreibvorgänge gleichmäßig auf die Partitions-Schlüsselwerte und somit auch über die physischen Partitionen verteilt. Sie können problemlos eine `SELECT` Anweisung für eine bestimmte Zeile und ein bestimmtes Datum ausführen, da Sie den Partitionsschlüsselwert für einen bestimmten `title` Wert berechnen können.

Um alle Zeilen für einen bestimmten Tag zu lesen, müssen Sie trotzdem `SELECT` jeden der (`2020-07-09`, `N`) Schlüssel (wo `N` 1—200) benötigen, und Ihre Anwendung muss dann alle Ergebnisse zusammenführen. Der Vorteil dabei ist, dass Sie so vermeiden, dass ein einzelner „Hot Partition“-Schlüsselwert die gesamte Workload übernimmt.

## Optimierung der Kosten von Amazon Keyspaces-Tabellen

In diesem Abschnitt werden bewährte Methoden zur Optimierung der Kosten für Ihre vorhandenen Amazon Keyspaces-Tabellen beschrieben. Sie sollten sich die folgenden Strategien zur Kostenoptimierung ansehen, um herauszufinden, welche davon Ihren Anforderungen am besten entspricht, und sie iterativ angehen. Jede Strategie bietet einen Überblick darüber, was sich auf Ihre Kosten auswirken könnte, wie Sie nach Möglichkeiten zur Kostenoptimierung suchen können, sowie

präskriptive Anleitungen zur Implementierung dieser bewährten Verfahren, um Ihnen beim Sparen zu helfen.

## Themen

- [Auswerten Ihrer Kosten auf Tabellenebene](#)
- [Evaluieren Sie den Kapazitätsmodus Ihrer Tabelle](#)
- [Evaluieren Sie die Application Auto Scaling Scaling-Einstellungen Ihrer Tabelle](#)
- [Identifizieren Sie Ihre ungenutzten Ressourcen](#)
- [Auswerten Ihrer Tabellennutzungsmuster](#)
- [Auswerten Ihrer bereitgestellten Kapazität, um eine angemessene Bereitstellung zu erzielen](#)

## Auswerten Ihrer Kosten auf Tabellenebene

Mit dem Cost Explorer Explorer-Tool im AWS Management Console können Sie die Kosten nach Typ aufgeschlüsselt anzeigen, z. B. Lese-, Schreib-, Speicher- und Backup-Gebühren. Sie können diese Kosten auch nach Zeitraum zusammengefasst sehen, beispielsweise die Kosten für einen Monat oder einen Tag.

Eine häufige Herausforderung bei Cost Explorer besteht darin, dass Sie die Kosten nur einer bestimmten Tabelle nicht einfach überprüfen können, da Sie mit Cost Explorer nicht nach den Kosten einer bestimmten Tabelle filtern oder gruppieren können. Sie können die metrische Größe der fakturierbaren Tabelle (Byte) jeder Tabelle in der Amazon Keyspaces-Konsole auf der Registerkarte Monitor der Tabelle einsehen. Wenn Sie mehr kostenbezogene Informationen pro Tabelle benötigen, erfahren Sie in diesem Abschnitt, wie Sie mithilfe von [Tagging](#) eine Kostenanalyse für einzelne Tabellen im Cost Explorer durchführen können.

## Themen

- [So zeigen Sie die Kosten einer einzelnen Amazon Keyspaces-Tabelle an](#)
- [Standardansicht von Cost Explorer](#)
- [So verwenden Sie Tabellen-Tags in Cost Explorer](#)

## So zeigen Sie die Kosten einer einzelnen Amazon Keyspaces-Tabelle an

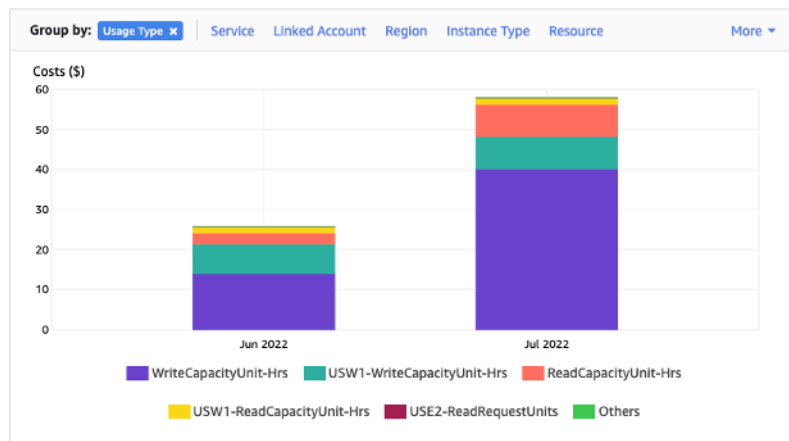
In der Konsole können Sie grundlegende Informationen zu einer Amazon Keyspaces-Tabelle einsehen, darunter das Primärschlüsselschema, die Größe der fakturierbaren Tabelle und

kapazitätsbezogene Kennzahlen. Sie können die Größe der Tabelle verwenden, um die monatlichen Speicherkosten für die Tabelle zu berechnen. Zum Beispiel 0,25\$ pro GB im Osten der USA (Nord-Virginia) AWS-Region.

Wenn die Tabelle den Modus für bereitgestellte Kapazität verwendet, werden auch die aktuellen Einstellungen für die Lesekapazitätseinheit (RCU) und die Schreibkapazitätseinheit (WCU) zurückgegeben. Sie können diese Informationen verwenden, um die aktuellen Lese- und Schreibkosten für die Tabelle zu berechnen. Beachten Sie, dass sich diese Kosten ändern können, insbesondere wenn Sie die Tabelle mit der automatischen Skalierung von Amazon Keyspaces konfiguriert haben.

## Standardansicht von Cost Explorer

Die Standardansicht im Cost Explorer enthält Diagramme, in denen die Kosten der verbrauchten Ressourcen, beispielsweise Durchsatz und Speicherplatz, dargestellt werden. Sie können diese Kosten nach Zeitraum gruppieren, z. B. nach Gesamtkosten nach Monat oder Tag. Die Kosten für Speicher, Lese- und Schreibvorgänge und andere Kategorien können ebenfalls aufgeschlüsselt und verglichen werden.



## So verwenden Sie Tabellen-Tags in Cost Explorer

Standardmäßig bietet der Cost Explorer keine Zusammenfassung der Kosten für eine bestimmte Tabelle, da er die Kosten mehrerer Tabellen zu einer Summe zusammenfasst. Sie können jedoch [AWS -Ressourcen-Tagging](#) verwenden, um jede Tabelle mit einem Metadaten-Tag zu identifizieren. Tags sind Schlüssel-Wert-Paare, die Sie für eine Vielzahl von Zwecken verwenden können, z. B. um alle Ressourcen zu identifizieren, die zu einem Projekt oder einer Abteilung gehören. Weitere Informationen finden Sie unter [the section called "Arbeiten mit Tags"](#).

In diesem Beispiel verwenden wir eine Tabelle mit dem Namen. MyTable

1. Setze ein Tag mit dem Schlüssel von `table_name` und dem Wert von `MyTable`
2. [Aktivieren Sie das Tag in Cost Explorer](#) und filtern Sie dann nach dem Tag-Wert, um einen besseren Einblick in die Kosten der einzelnen Tabellen zu erhalten.

### Note

Es kann ein oder zwei Tage dauern, bis das Tag in Cost Explorer angezeigt wird.

Sie können Metadaten-Tags selbst in der Konsole oder programmgesteuert mit CQL, dem oder dem AWS CLI SDK festlegen. AWS Überlegen Sie sich, im Rahmen des neuen Tabellenerstellungsprozesses Ihrer Organisation die Festlegung eines `table_name`-Tags zu erfordern. Weitere Informationen finden Sie unter [the section called “Kostenzuordnungsberichte für Amazon Keyspaces”](#).

## Evaluieren Sie den Kapazitätsmodus Ihrer Tabelle

Dieser Abschnitt bietet einen Überblick darüber, wie Sie den geeigneten Kapazitätsmodus für Ihre Amazon Keyspaces-Tabelle auswählen. Jeder Modus ist auf die Anforderungen einer anderen Workload abgestimmt, sowohl was die Reaktionsfähigkeit auf Durchsatzänderungen, als auch was die Abrechnung der Nutzung anbelangt. Sie müssen diese Faktoren abwägen, wenn Sie Ihre Entscheidung treffen.

### Themen

- [Verfügbare Tabellenkapazitätsmodi](#)
- [Fälle, in denen der On-Demand-Modus ausgewählt werden sollte](#)
- [Fälle, in denen der Modus bereitgestellter Kapazität ausgewählt werden sollte](#)
- [Zusätzliche Faktoren, die bei der Auswahl eines Tabellenkapazitätsmodus zu berücksichtigen sind](#)

## Verfügbare Tabellenkapazitätsmodi

Wenn Sie eine Amazon Keyspaces-Tabelle erstellen, müssen Sie entweder den Modus „On-Demand-Kapazität“ oder „Bereitgestellte Kapazität“ wählen. Weitere Informationen finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).

### On-Demand-Kapazitätsmodus

Der On-Demand-Kapazitätsmodus ist so konzipiert, dass Sie die Kapazität Ihrer Amazon Keyspaces-Tabelle nicht mehr planen oder bereitstellen müssen. In diesem Modus bearbeitet Ihre Tabelle Anfragen sofort, ohne dass Ressourcen nach oben oder unten skaliert werden müssen (bis zum Doppelten des vorherigen Spitzendurchsatzes der Tabelle).

On-Demand-Tabellen werden abgerechnet, indem die Anzahl der tatsächlichen Anfragen mit der Tabelle verglichen wird. Sie zahlen also nur für das, was Sie tatsächlich nutzen, und nicht für das, was bereitgestellt wurde.

### Modus bereitgestellter Kapazität

Der Modus für bereitgestellte Kapazität ist ein traditionelleres Modell, bei dem Sie entweder direkt oder mit Hilfe von Application Auto Scaling definieren können, wie viel Kapazität die Tabelle für Anfragen zur Verfügung hat. Da zu jedem Zeitpunkt eine bestimmte Kapazität für die Tabelle bereitgestellt wird, basiert die Abrechnung auf der bereitgestellten Kapazität und nicht auf der Anzahl der Anforderungen. Eine Überschreitung der zugewiesenen Kapazität kann auch dazu führen, dass die Tabelle Anfragen ablehnt und die Benutzererfahrung der Benutzer Ihrer Anwendung beeinträchtigt.

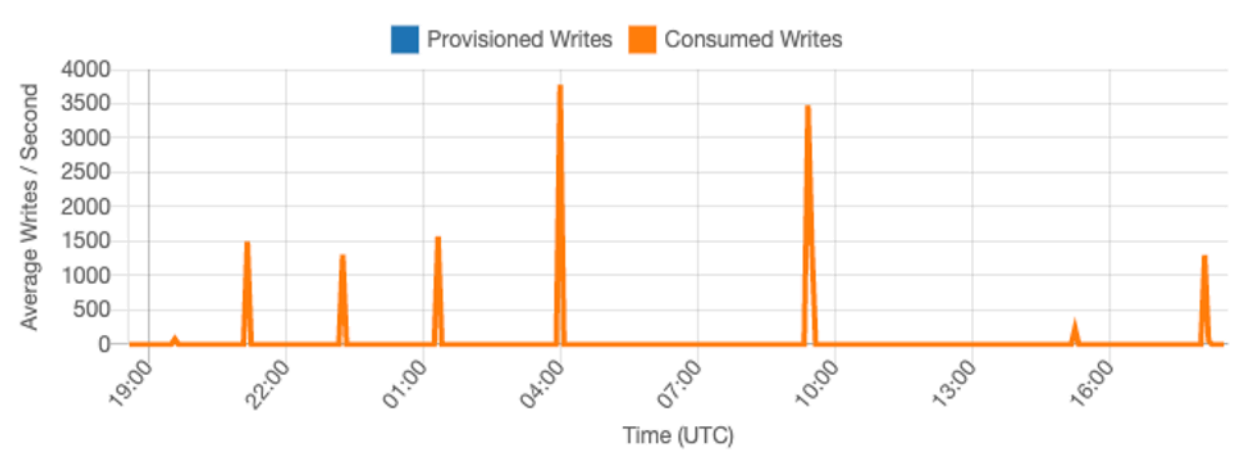
Der Modus „Bereitgestellte Kapazität“ erfordert ein ausgewogenes Verhältnis zwischen der Nichtbereitstellung und der Unterbereitstellung der Tabelle, um sowohl das geringe Auftreten von Fehlern bei unzureichender Durchsatzkapazität als auch die Optimierung der Kosten zu erreichen.

### Fälle, in denen der On-Demand-Modus ausgewählt werden sollte

Bei der Kostenoptimierung ist der On-Demand-Modus die beste Wahl, wenn Sie eine unvorhersehbare Arbeitslast haben, die der in der folgenden Grafik dargestellten ähnelt.

Diese Faktoren tragen zu dieser Art von Arbeitslast bei:

- Unvorhersehbares Timing der Anforderungen (dadurch Datenverkehrsspitzen)
- Variables Volumen der Anforderungen (aufgrund von Batch-Workloads)
- Sinkt in einer bestimmten Stunde auf Null oder unter 18% des Spitzenwerts (aufgrund von Entwicklungs- oder Testumgebungen)



Bei Workloads mit den oben genannten Merkmalen kann die Verwendung von Application Auto Scaling, um genügend Kapazität aufrechtzuerhalten, damit die Tabelle auf Datenverkehrsspitzen reagieren kann, zu unerwünschten Ergebnissen führen. Entweder ist die Tabelle überdimensioniert und kostet mehr als nötig, oder die Tabelle ist zu wenig bereitgestellt und Anfragen führen zu unnötigen Durchsatzfehlern bei niedriger Kapazität. In solchen Fällen sind On-Demand-Tabellen die bessere Wahl.

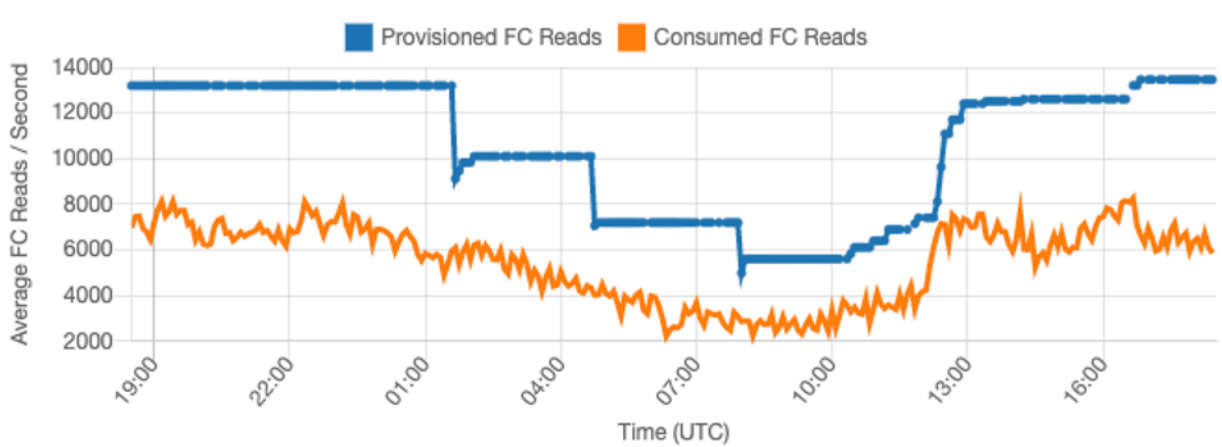
Da On-Demand-Tabellen auf Anfrage abgerechnet werden, müssen Sie auf Tabellenebene nichts weiter tun, um die Kosten zu optimieren. Sie sollten Ihre On-Demand-Tabellen regelmäßig überprüfen, um sicherzustellen, dass der Workload weiterhin die oben genannten Merkmale aufweist. Wenn sich die Arbeitslast stabilisiert hat, sollten Sie erwägen, zum Bereitstellungsmodus zu wechseln, um die Kostenoptimierung aufrechtzuerhalten.

Fälle, in denen der Modus bereitgestellter Kapazität ausgewählt werden sollte

Ein idealer Workload für den Modus mit bereitgestellter Kapazität ist ein Workload mit einem besser vorhersehbaren Nutzungsmuster, wie in der folgenden Grafik dargestellt.

Die folgenden Faktoren tragen zu einer vorhersehbaren Arbeitslast bei:

- Vorhersagbarer/zyklischer Datenverkehr für eine bestimmte Stunde oder einen bestimmten Tag
- Begrenzte kurzfristige Datenverkehrsspitzen



Da das Datenverkehrsvolumen innerhalb einer bestimmten Zeit oder eines bestimmten Tages stabiler ist, können Sie die bereitgestellte Kapazität relativ nahe an der tatsächlich verbrauchten Kapazität der Tabelle festlegen. Bei der Kostenoptimierung einer Tabelle mit bereitgestellter Kapazität geht es letztlich darum, die bereitgestellte Kapazität (blaue Linie) so nah wie möglich an die verbrauchte Kapazität (orange Linie) heranzuführen, ohne die Anzahl der `ThrottledRequests` Ereignisse in der Tabelle zu erhöhen. Der Abstand zwischen den beiden Zeilen ist sowohl verschwendete Kapazität als auch eine Versicherung gegen eine schlechte Benutzererfahrung aufgrund unzureichender Durchsatzkapazitätsfehler.

Amazon Keyspaces bietet Application Auto Scaling für bereitgestellte Kapazitätstabellen, das diese automatisch in Ihrem Namen ausgleicht. Sie können Ihre verbrauchte Kapazität im Laufe des Tages verfolgen und die bereitgestellte Kapazität der Tabelle anhand einer Handvoll Variablen konfigurieren.

### Einheiten für minimale Kapazität

Sie können die Mindestkapazität einer Tabelle festlegen, um das Auftreten von Fehlern bei unzureichender Durchsatzkapazität zu begrenzen. Die Kosten der Tabelle werden dadurch jedoch nicht gesenkt. Wenn Ihre Tabelle Perioden mit geringer Auslastung aufweist, gefolgt von einem plötzlichen Anstieg hoher Auslastung, kann die Festlegung des Minimums verhindern, dass Application Auto Scaling die Tabellenkapazität zu niedrig festlegt.

### Einheiten für maximale Kapazität

Sie können die maximale Kapazität einer Tabelle festlegen, um zu verhindern, dass eine Tabelle stärker als beabsichtigt skaliert wird. Erwägen Sie, ein Maximum für Entwicklungs- oder Testtabellen festzulegen, wenn umfangreiche Auslastungstests nicht erwünscht sind. Sie können für jede Tabelle einen Höchstwert festlegen. Achten Sie jedoch darauf, diese Einstellung regelmäßig mit

der Tabellenbasislinie abzugleichen, wenn Sie sie in der Produktion verwenden, um versehentliche Fehler bei unzureichender Durchsatzkapazität zu vermeiden.

## Zielauslastung

Die Festlegung einer Zielauslastung der Tabelle ist das primäre Mittel zur Kostenoptimierung für eine Tabelle mit bereitgestellter Kapazität. Wenn Sie hier einen niedrigeren Prozentwert angeben, erhöht sich die Überbelegung der Tabelle, was die Kosten erhöht, aber das Risiko von Fehlern bei unzureichender Durchsatzkapazität verringert. Wenn Sie einen höheren Prozentwert festlegen, verringert sich die Überbelegung der Tabelle, erhöht sich jedoch das Risiko von Fehlern bei unzureichender Durchsatzkapazität.

## Zusätzliche Faktoren, die bei der Auswahl eines Tabellenkapazitätsmodus zu berücksichtigen sind

Bei der Entscheidung zwischen den beiden Kapazitätsmodi sollten einige zusätzliche Faktoren berücksichtigt werden.

Berücksichtigen Sie bei der Entscheidung zwischen den beiden Tabellenmodi, wie stark sich dieser zusätzliche discount auf die Kosten des Tisches auswirkt. In vielen Fällen kann selbst eine relativ unvorhersehbare Arbeitslast kostengünstiger sein, wenn sie auf einer Tabelle mit überdimensionierter Kapazität und reservierter Kapazität ausgeführt wird.

## Verbessern der Vorhersehbarkeit Ihrer Workload

In manchen Situationen kann ein Workload scheinbar sowohl ein vorhersehbares als auch ein unvorhersehbares Muster aufweisen. Dies kann zwar mit einer On-Demand-Tabelle problemlos unterstützt werden, aber die Kosten werden wahrscheinlich niedriger sein, wenn die unvorhersehbaren Muster in der Arbeitslast verbessert werden können.

Eine der häufigsten Ursachen für diese Muster sind Batch-Importe. Diese Art von Verkehr kann die Basiskapazität der Tabelle oft so stark überschreiten, dass bei der Ausführung Fehler mit unzureichender Durchsatzkapazität auftreten würden. Um eine Workload wie diese in einer Tabelle mit bereitgestellter Kapazität auszuführen, sollten Sie die folgenden Optionen in Betracht ziehen:

- Wenn der Batch zu geplanten Zeiten ausgeführt wird, können Sie eine Erhöhung der Kapazität Ihrer Anwendung für die auto Skalierung planen, bevor sie ausgeführt wird.
- Wenn der Batch zufällig ausgeführt wird, sollten Sie versuchen, die für die Ausführung benötigte Zeit zu verlängern, anstatt ihn so schnell wie möglich auszuführen.



- Fügen Sie dem Import eine Anlaufphase hinzu, in der die Geschwindigkeit des Imports gering beginnt, aber über einige Minuten langsam erhöht wird, bis Application Auto Scaling die Möglichkeit hatte, mit der Anpassung der Tabellenkapazität zu beginnen.

## Evaluieren Sie die Application Auto Scaling Scaling-Einstellungen Ihrer Tabelle

Dieser Abschnitt bietet einen Überblick darüber, wie Sie die Application Auto Scaling Scaling-Einstellungen in Ihren Amazon Keyspaces-Tabellen auswerten können. [Amazon Keyspaces Application Auto Scaling](#) ist eine Funktion, die den Tabellendurchsatz auf der Grundlage Ihres Anwendungsdatenverkehrs und Ihrer Zielnutzungsmetrik verwaltet. Dadurch wird sichergestellt, dass Ihre Tabellen über die für Ihre Anwendungsmuster erforderliche Kapazität verfügen.

Der Application Auto Scaling Scaling-Dienst überwacht Ihre aktuelle Tabellenauslastung und vergleicht sie mit dem Zielnutzungswert: `TargetValue`. Er benachrichtigt Sie, wenn es an der Zeit ist, die zugewiesene Kapazität zu erhöhen oder zu verringern.

### Themen

- [Grundlegendes zu Ihren Application Auto Scaling Scaling-Einstellungen](#)
- [So ermitteln Sie Tabellen mit geringer Zielauslastung \(<= 50 %\)](#)
- [So bewältigen Sie Workloads mit saisonalen Schwankungen](#)
- [So bewältigen Sie stark schwankende Workloads mit unbekanntem Mustern](#)
- [So bewältigen Sie Workloads mit verknüpften Anwendungen](#)

## Grundlegendes zu Ihren Application Auto Scaling Scaling-Einstellungen

Die Festlegung des richtigen Werts für die Zielauslastung, den ersten Schritt und die Endwerte erfordert die Beteiligung Ihres Operations-Teams. Auf diese Weise können Sie die Werte auf der Grundlage der historischen Anwendungsnutzung, die zum Auslösen der Application Auto Scaling Scaling-Richtlinien verwendet wird, richtig definieren. Das Auslastungsziel ist der Prozentsatz Ihrer Gesamtkapazität, der während eines bestimmten Zeitraums erreicht werden muss, bevor die Application Auto Scaling Scaling-Regeln gelten.

Wenn Sie ein hohes Auslastungsziel festlegen (ein Ziel von etwa 90%), bedeutet dies, dass Ihr Traffic für einen bestimmten Zeitraum höher als 90% sein muss, bevor Application Auto Scaling aktiviert

wird. Ein hohes Auslastungsziel sollten Sie nur verwenden, wenn Ihre Anwendung sehr konstant arbeitet und keine Datenverkehrsspitzen verzeichnet.

Wenn Sie eine sehr niedrige Auslastung festlegen (ein Ziel von weniger als 50%), bedeutet dies, dass Ihre Anwendung 50% der bereitgestellten Kapazität erreichen müsste, bevor sie eine Application Auto Scaling Scaling-Richtlinie auslöst. Sofern Ihr Anwendungsdatenverkehr nicht extrem schnell zunimmt, führt dies in der Regel zu ungenutzter Kapazität und einer Ressourcenverschwendung.

So ermitteln Sie Tabellen mit geringer Zielauslastung ( $\leq 50\%$ )

Sie können entweder das AWS CLI oder verwenden AWS Management Console, um die Auto Scaling-Richtlinien TargetValues für Ihre Anwendung in Ihren Amazon Keyspaces-Ressourcen zu überwachen und zu identifizieren:

## AWS CLI

1. Verwenden Sie den folgenden Befehl, um die gesamte Ressourcenliste zurückzugeben:

```
aws application-autoscaling describe-scaling-policies --service-namespace
cassandra
```

Dieser Befehl gibt die gesamte Liste der Application Auto Scaling Scaling-Richtlinien zurück, die für jede Amazon Keyspaces-Ressource ausgestellt wurden. Wenn Sie nur die Ressourcen aus einer bestimmten Tabelle abrufen möchten, können Sie den `-resource-id` parameter hinzufügen. Beispielsweise:

```
aws application-autoscaling describe-scaling-policies --service-namespace
cassandra --resource-id "keyspace/keyspace-name/table/table-name"
```

2. Geben Sie nur die Auto Scaling-Richtlinien für eine bestimmte Tabelle zurück, indem Sie den folgenden Befehl ausführen

```
aws application-autoscaling describe-scaling-policies --service-namespace
cassandra --resource-id "keyspace/keyspace-name/table/table-name"
```

Die Werte für die Application Auto Scaling Scaling-Richtlinien sind unten hervorgehoben. Sie müssen sicherstellen, dass der Zielwert über 50% liegt, um eine übermäßige Bereitstellung zu vermeiden. Das Ergebnis sollte etwa wie folgt aussehen:

```
{
```

```

"ScalingPolicies": [
 {
 "PolicyARN": "arn:aws:autoscaling:<region>:<account-
id>:scalingPolicy:<uuid>:resource/keyspaces/table/table-name-scaling-policy",
 "PolicyName": "$<full-gsi-name>",
 "ServiceNamespace": "cassandra",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:index:WriteCapacityUnits",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 70.0,
 "PredefinedMetricSpecification": {
 "PredefinedMetricType": "KeyspacesWriteCapacityUtilization"
 }
 },
 "Alarms": [
 ...
],
 "CreationTime": "2022-03-04T16:23:48.641000+10:00"
 },
 {
 "PolicyARN": "arn:aws:autoscaling:<region>:<account-
id>:scalingPolicy:<uuid>:resource/keyspaces/table/table-name/index/<index-
name>:policyName/$<full-gsi-name>-scaling-policy",
 "PolicyName": "$<full-table-name>",
 "ServiceNamespace": "cassandra",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:index:ReadCapacityUnits",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 70.0,
 "PredefinedMetricSpecification": {
 "PredefinedMetricType": "CassandraReadCapacityUtilization"
 }
 },
 "Alarms": [
 ...
],
 "CreationTime": "2022-03-04T16:23:47.820000+10:00"
 }
]
}

```

## AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und navigieren Sie zur CloudWatch Serviceseite unter [Erste Schritte mit dem](#). AWS Management Console Wählen Sie bei AWS-Region Bedarf das entsprechende aus.
2. Wählen Sie in der linken Navigationsleiste die Option Tables (Tabellen) aus. Wählen Sie auf der Seite Tables (Tabellen) den Namen der Tabelle aus.
3. Überprüfen Sie auf der Seite Tabellendetails auf der Registerkarte Kapazität die Einstellungen für Application Auto Scaling Ihrer Tabelle.

Bei Zielauslastungswerten von maximal 50 % sollten Sie Ihre Tabellenauslastungsmetriken prüfen, um festzustellen, ob eine [zu geringe oder eine übermäßige Bereitstellung](#) vorliegt.

## So bewältigen Sie Workloads mit saisonalen Schwankungen

Stellen Sie sich folgendes Szenario vor: Ihre Anwendung läuft die meiste Zeit unter einem minimalen Durchschnittswert, doch das Auslastungsziel ist niedrig. Somit kann Ihre Anwendung schnell auf Ereignisse reagieren, die zu bestimmten Tageszeiten auftreten, Sie verfügen über genügend Kapazität und es kommt nicht zu Drosselungen. Ein solches Szenario kommt häufig vor, wenn eine Anwendung während der normalen Bürozeiten (9 bis 17 Uhr) sehr stark ausgelastet ist, außerhalb der Geschäftszeiten jedoch auf niedrigem Niveau läuft. Da einige Benutzer vor 9 Uhr eine Verbindung herstellen, verwendet die Anwendung diesen niedrigen Schwellenwert, um schnell hochzufahren und zu Spitzenzeiten die erforderliche Kapazität zu erreichen.

Dieses Szenario könnte wie folgt aussehen:

- Zwischen 17 Uhr und 9 Uhr bleiben die ConsumedWriteCapacityUnits-Einheiten zwischen 90 und 100.
- Benutzer beginnen vor 9 Uhr, eine Verbindung zu der Anwendung herzustellen, und die Kapazitätseinheiten steigen erheblich an (der maximale Wert, den Sie gesehen haben, beträgt 1 500 WCU).
- Im Durchschnitt variiert Ihre Anwendungsnutzung während der Arbeitszeit zwischen 800 und 1 200.

Wenn das vorherige Szenario auf Ihre Anwendung zutrifft, sollten Sie die [Application Auto Scaling nach Zeitplan](#) in Betracht ziehen, bei der in Ihrer Tabelle immer noch eine Regel für Application Auto Scaling konfiguriert sein könnte, aber mit einer weniger aggressiven Zielauslastung, die die zusätzliche Kapazität nur in den von Ihnen benötigten Intervallen bereitstellt.

Mit dem können Sie die AWS CLI folgenden Schritte ausführen, um eine geplante Auto Scaling-Regel zu erstellen, die auf der Grundlage der Tageszeit und des Wochentags ausgeführt wird.

1. Registrieren Sie Ihre Amazon Keyspaces-Tabelle als skalierbares Ziel bei Application Auto Scaling. Ein skalierbares Ziel ist eine Ressource, die Application Auto Scaling auf- und abskalieren kann.

```
aws application-autoscaling register-scalable-target \
 --service-namespace cassandra \
 --scalable-dimension cassandra:table:WriteCapacityUnits \
 --resource-id keyspace/keyspace-name/table/table-name \
 --min-capacity 90 \
 --max-capacity 1500
```

2. Richten Sie geplante Aktionen entsprechend Ihren Anforderungen ein.

Sie benötigen zwei Regeln, um das Szenario abzudecken: eine für die Hochskalierung und eine weitere für die Verkleinerung. Die erste Regel zum Hochskalieren der geplanten Aktion wird im folgenden Beispiel gezeigt.

```
aws application-autoscaling put-scheduled-action \
 --service-namespace cassandra \
 --scalable-dimension cassandra:table:WriteCapacityUnits \
 --resource-id keyspace/keyspace-name/table/table-name \
 --scheduled-action-name my-8-5-scheduled-action \
 --scalable-target-action MinCapacity=800,MaxCapacity=1500 \
 --schedule "cron(45 8 ? * MON-FRI *)" \
 --timezone "Australia/Brisbane"
```

In diesem Beispiel wird die zweite Regel zum Verkleinern der geplanten Aktion gezeigt.

```
aws application-autoscaling put-scheduled-action \
 --service-namespace cassandra \
 --scalable-dimension cassandra:table:WriteCapacityUnits \
 --resource-id keyspace/keyspace-name/table/table-name \
 --scheduled-action-name my-5-8-scheduled-down-action \
 --scalable-target-action MinCapacity=90,MaxCapacity=1500 \
 --schedule "cron(15 17 ? * MON-FRI *)" \
 --timezone "Australia/Brisbane"
```

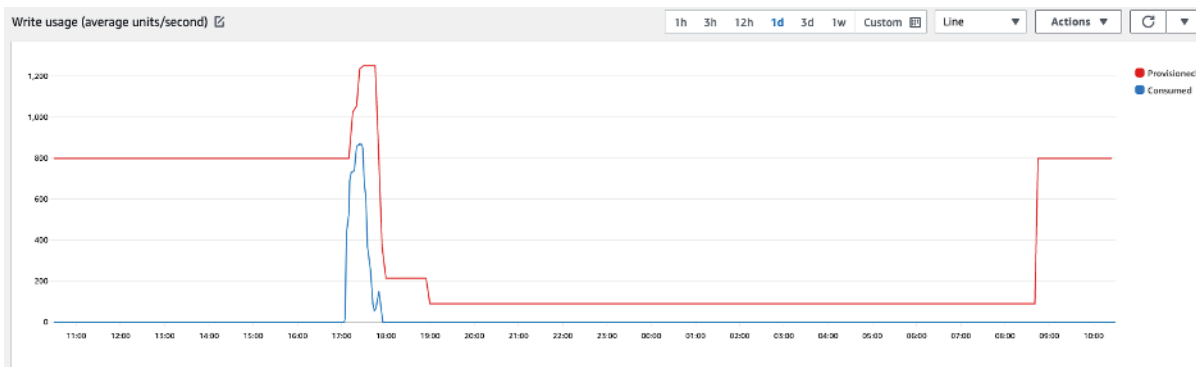
3. Führen Sie den folgenden Befehl aus, um zu bestätigen, dass beide Regeln aktiviert wurden:

```
aws application-autoscaling describe-scheduled-actions --service-namespace
cassandra
```

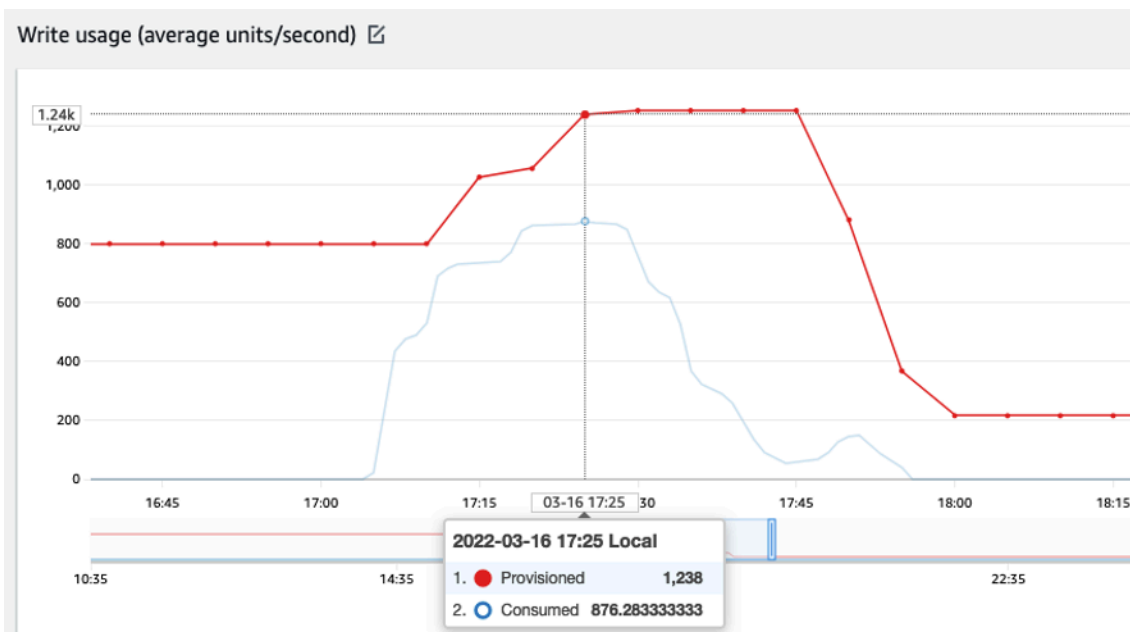
Das Ergebnis sollte ungefähr wie folgt aussehen:

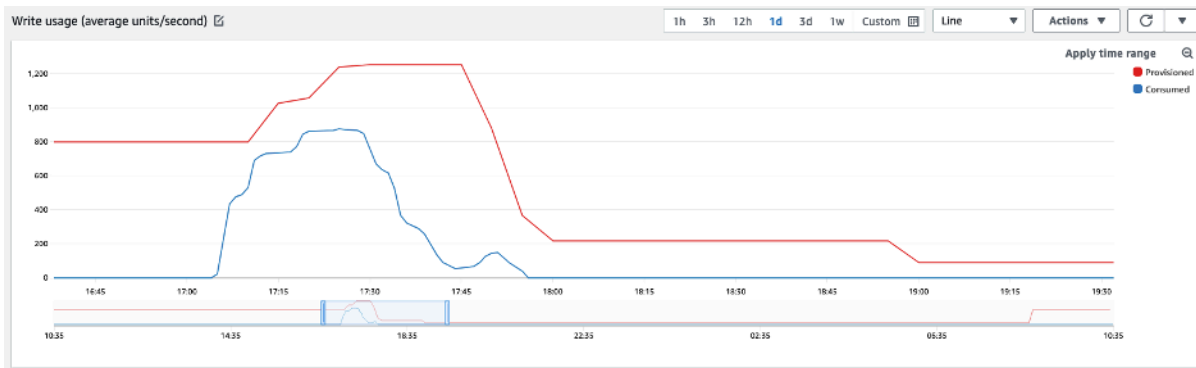
```
{
 "ScheduledActions": [
 {
 "ScheduledActionName": "my-5-8-scheduled-down-action",
 "ScheduledActionARN":
"arn:aws:autoscaling:<region>:<account>:scheduledAction:<uuid>:resource/keyspaces/
table/table-name:scheduledActionName/my-5-8-scheduled-down-action",
 "ServiceNamespace": "cassandra",
 "Schedule": "cron(15 17 ? * MON-FRI *)",
 "Timezone": "Australia/Brisbane",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:table:WriteCapacityUnits",
 "ScalableTargetAction": {
 "MinCapacity": 90,
 "MaxCapacity": 1500
 },
 "CreationTime": "2022-03-15T17:30:25.100000+10:00"
 },
 {
 "ScheduledActionName": "my-8-5-scheduled-action",
 "ScheduledActionARN":
"arn:aws:autoscaling:<region>:<account>:scheduledAction:<uuid>:resource/keyspaces/
table/table-name:scheduledActionName/my-8-5-scheduled-action",
 "ServiceNamespace": "cassandra",
 "Schedule": "cron(45 8 ? * MON-FRI *)",
 "Timezone": "Australia/Brisbane",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:table:WriteCapacityUnits",
 "ScalableTargetAction": {
 "MinCapacity": 800,
 "MaxCapacity": 1500
 },
 "CreationTime": "2022-03-15T17:28:57.816000+10:00"
 }
]
}
```

Im folgenden Bild ist ein Workload zu sehen, bei dem immer die Zielauslastung von 70 % beibehalten wird. Beachten Sie, dass die Auto Scaling-Regeln immer noch gelten und der Durchsatz nicht reduziert wird.



Wenn wir die Ansicht vergrößern, sehen wir, dass es in der Anwendung einen Spitzenwert gegeben hat, der den 70%-Schwellenwert für Auto Scaling ausgelöst hat, sodass Auto Scaling aktiviert und die für die Tabelle benötigte zusätzliche Kapazität bereitgestellt wurde. Die geplante Auto Scaling-Aktion wirkt sich auf Höchst- und Minimalwerte aus, und es liegt in Ihrer Verantwortung, sie einzurichten.





## So bewältigen Sie stark schwankende Workloads mit unbekanntem Mustern

In diesem Szenario verwendet die Anwendung ein sehr niedriges Auslastungsziel, da Sie die Anwendungsmuster noch nicht kennen und Sie sicherstellen möchten, dass es bei Ihrem Workload nicht zu Durchsatzfehlern mit niedriger Kapazität kommt.

Sie sollten stattdessen die Verwendung des [On-Demand-Kapazitätsmodus](#) in Betracht ziehen. On-Demand-Tabellen eignen sich perfekt für stark schwankende Workloads, deren Datenverkehrsmuster Sie nicht kennen. Im On-Demand-Kapazitätsmodus zahlen Sie pro Anforderung für die Lese- und Schreibvorgänge, die Ihre Anwendung in Ihren Tabellen ausführt. Sie müssen nicht angeben, wie viel Lese- und Schreibdurchsatz Sie von Ihrer Anwendung erwarten, da Amazon Keyspaces Ihre Workloads sofort berücksichtigt, wenn sie hoch- oder herunterfahren.

## So bewältigen Sie Workloads mit verknüpften Anwendungen

In diesem Szenario hängt die Anwendung von anderen Systemen ab. Dies kann beispielsweise in Batchverarbeitungsszenarien der Fall sein, bei denen es abhängig von Ereignissen in der Anwendungslogik zu starken Datenverkehrsspitzen kommen kann.

Erwägen Sie die Entwicklung einer benutzerdefinierten Logik zur auto-scaling von Anwendungen, die auf Ereignisse reagiert, bei denen Sie die Tabellenkapazität erhöhen können TargetValues und die Ihren spezifischen Anforderungen entspricht. Sie könnten von einer Kombination von AWS Diensten wie  $\wedge$  Amazon EventBridge und Step Functions profitieren und diese nutzen, um auf Ihre spezifischen Anwendungsanforderungen zu reagieren.

## Identifizieren Sie Ihre ungenutzten Ressourcen

Dieser Abschnitt gibt einen Überblick darüber, wie Sie Ihre nicht verwendeten Ressourcen regelmäßig bewerten können. Wenn sich Ihre Anwendungsanforderungen weiterentwickeln, sollten Sie sicherstellen, dass keine Ressourcen ungenutzt bleiben und zu unnötigen Amazon Keyspaces-



Kosten beitragen. Die unten beschriebenen Verfahren verwenden CloudWatch Amazon-Metriken, um ungenutzte Ressourcen zu identifizieren und Maßnahmen zur Kostensenkung zu ergreifen.

Sie können Amazon Keyspaces mithilfe von Amazon Keyspaces überwatchen CloudWatch, das Rohdaten von Amazon Keyspaces sammelt und zu lesbaren, nahezu in Echtzeit verfügbaren Metriken verarbeitet. Diese Statistiken werden eine gewisse Zeit aufbewahrt, damit Sie zum besseren Verständnis Ihrer Nutzung Verlaufsdaten zur Verfügung haben. Standardmäßig werden Amazon Keyspaces-Metriken CloudWatch automatisch an gesendet. Weitere Informationen finden Sie unter [Was ist Amazon CloudWatch?](#) und [Aufbewahrung von Kennzahlen](#) im CloudWatch Amazon-Benutzerhandbuch.

## Themen

- [So ermitteln Sie nicht verwendete Ressourcen](#)
- [Ermitteln von nicht verwendeten Tabellenressourcen](#)
- [Bereinigen von nicht verwendeten Tabellenressourcen](#)
- [Bereinigen ungenutzter point-in-time Wiederherstellungs-Backups \(PITR\)](#)

## So ermitteln Sie nicht verwendete Ressourcen

Um ungenutzte Tabellen zu identifizieren, können Sie sich die folgenden CloudWatch Kennzahlen über einen Zeitraum von 30 Tagen ansehen, um festzustellen, ob in einer bestimmten Tabelle aktive Lese- oder Schreibvorgänge vorhanden sind:

### **ConsumedReadCapacityUnits**

Die Anzahl der in einem bestimmten Zeitraum verbrauchten Lesekapazitätseinheiten, um nachverfolgen zu können, wie viel Kapazität Sie genutzt haben. Sie können die gesamte verbrauchte Lesekapazität für eine Tabelle abrufen.

### **ConsumedWriteCapacityUnits**

Die Anzahl der in einem bestimmten Zeitraum verbrauchten Schreibkapazitätseinheiten, um nachverfolgen zu können, wie viel Kapazität Sie genutzt haben. Sie können die gesamte verbrauchte Schreibkapazität für eine Tabelle abrufen.

## Ermitteln von nicht verwendeten Tabellenressourcen

Amazon CloudWatch ist ein Überwachungs- und Beobachtbarkeitservice, der die Amazon Keyspaces-Tabellenmetriken bereitstellt, anhand derer Sie ungenutzte Ressourcen identifizieren

können. CloudWatch Metriken können sowohl über AWS Management Console als auch über die eingesehen werden. AWS Command Line Interface

## AWS Command Line Interface

Um die Metriken Ihrer Tabellen über anzuzeigen AWS Command Line Interface, können Sie die folgenden Befehle verwenden.

1. Werten Sie zunächst die Lesevorgänge Ihrer Tabelle aus:

### Note

Wenn der Tabellename in Ihrem Konto nicht eindeutig ist, müssen Sie auch den Namen des Schlüsselraums angeben.

```
aws cloudwatch get-metric-statistics --metric-name
ConsumedReadCapacityUnits --start-time <start-time> --end-time <end-
time> --period <period> --namespace AWS/Cassandra --statistics Sum --
dimensions Name=TableName,Value=<table-name>
```

Damit Tabellen nicht fälschlicherweise als nicht verwendet ermittelt werden, sollten Sie die Metriken über einen längeren Zeitraum auswerten. Wählen Sie einen geeigneten Start- und Endzeitbereich, z. B. 30 Tage, und einen geeigneten Zeitraum, z. B. 86400.

In den zurückgegebenen Daten zeigt eine Summe von mehr als 0 an, dass die auszuwertende Tabelle während dieses Zeitraums Lesedatenverkehr empfangen hat.

Das folgende Ergebnis zeigt eine Tabelle, die im ausgewerteten Zeitraum Lesedatenverkehr empfangen hat:

```
{
 "Timestamp": "2022-08-25T19:40:00Z",
 "Sum": 36023355.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-12T19:40:00Z",
 "Sum": 38025777.5,
 "Unit": "Count"
}
```

```
},
```

Das folgende Ergebnis zeigt eine Tabelle, die im ausgewerteten Zeitraum keinen Lesedatenverkehr empfangen hat:

```
{
 "Timestamp": "2022-08-01T19:50:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-20T19:50:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
```

## 2. Werten Sie als Nächstes die Schreibvorgänge Ihrer Tabelle aus:

```
aws cloudwatch get-metric-statistics --metric-name
ConsumedWriteCapacityUnits --start-time <start-time> --end-time <end-
time> --period <period> --namespace AWS/Cassandra --statistics Sum --
dimensions Name=TableName,Value=<table-name>
```

Damit Tabellen nicht fälschlicherweise als nicht verwendet ermittelt werden, sollten Sie die Metriken über einen längeren Zeitraum auswerten. Wählen Sie einen geeigneten Startzeit- und Endzeitbereich, beispielsweise 30 Tage, und einen geeigneten Zeitraum, wie z. B. 86400.

In den zurückgegebenen Daten zeigt eine Summe von mehr als 0 an, dass die auszuwertende Tabelle während dieses Zeitraums Lesedatenverkehr empfangen hat.

Das folgende Ergebnis zeigt eine Tabelle, die im ausgewerteten Zeitraum Schreibdatenverkehr empfangen hat:

```
{
 "Timestamp": "2022-08-19T20:15:00Z",
 "Sum": 41014457.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-18T20:15:00Z",
```

```
 "Sum": 40048531.0,
 "Unit": "Count"
 },
```

Das folgende Ergebnis zeigt eine Tabelle, die im ausgewerteten Zeitraum keinen Schreibdatenverkehr empfangen hat:

```
{
 "Timestamp": "2022-07-31T20:15:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-19T20:15:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
```

## AWS Management Console

Mit den folgenden Schritten können Sie Ihre Ressourcennutzung anhand der bewerten. AWS Management Console

1. Melden Sie sich bei <https://console.aws.amazon.com/cloudwatch/> an AWS Management Console und rufen Sie die CloudWatch Serviceseite auf. Wählen Sie bei Bedarf oben rechts AWS-Region in der Konsole die entsprechende Option aus.
2. Suchen Sie in der linken Navigationsleiste den Abschnitt Metriken und wählen Sie Alle Metriken aus.
3. Die obige Aktion öffnet ein Dashboard mit zwei Bereichen. Im oberen Bereich sehen Sie die aktuellen Kennzahlen grafisch dargestellt. Unten können Sie die Metriken auswählen, die grafisch dargestellt werden können. Wählen Sie im unteren Bereich Amazon Keyspaces aus.
4. Wählen Sie im Auswahlfeld für Amazon Keyspaces-Metriken die Kategorie Tabellenmetriken aus, um die Metriken für Ihre Tabellen in der aktuellen Region anzuzeigen.
5. Identifizieren Sie Ihren Tabellennamen, indem Sie im Menü nach unten scrollen und dann die Metriken ConsumedReadCapacityUnits und ConsumedWriteCapacityUnits für Ihre Tabelle auswählen.
6. Wählen Sie den Tab Graphed Metrics (2) und passen Sie die Spalte Statistik auf Summe an.

- Um zu vermeiden, dass eine Tabelle fälschlicherweise als unbenutzt identifiziert wird, werten Sie die Tabellenmetriken über einen längeren Zeitraum aus. Wählen Sie oben im Grafikfenster einen geeigneten Zeitraum, z. B. einen Monat, für die Auswertung Ihrer Tabelle aus. Wählen Sie Benutzerdefiniert, wählen Sie im Dropdownmenü die Option 1 Monat aus und wählen Sie Anwenden aus.
- Werten Sie die grafisch dargestellten Metriken für Ihre Tabelle aus, um festzustellen, ob die Tabelle genutzt wird. Metriken, die über 0 hinausgehen, weisen darauf hin, dass eine Tabelle während des ausgewerteten Zeitraums genutzt wurde. Ein flaches Diagramm bei 0 für Lese- und Schreibvorgänge zeigt an, dass eine Tabelle unbenutzt ist.

## Bereinigen von nicht verwendeten Tabellenressourcen

Wenn Sie nicht verwendete Tabellenressourcen ermittelt haben, können Sie die laufenden Kosten für diese Ressourcen auf folgende Weise reduzieren.

### Note

Wenn Sie eine nicht verwendete Tabelle ermittelt haben, die jedoch verfügbar bleiben soll, falls in Zukunft darauf zugegriffen werden muss, sollten Sie eine Umstellung auf den On-Demand-Modus in Betracht ziehen. Andernfalls können Sie erwägen, die Tabelle zu löschen.

## Kapazitätsmodi

Amazon Keyspaces berechnet Gebühren für das Lesen, Schreiben und Speichern von Daten in Ihren Amazon Keyspaces-Tabellen.

Amazon Keyspaces bietet [zwei Kapazitätsmodi](#) mit spezifischen Abrechnungsoptionen für die Verarbeitung von Lese- und Schreibvorgängen in Ihren Tabellen: auf Abruf und bereitgestellt. Der Lese-/Schreibkapazitätsmodus steuert, wie Ihnen der Lese- und Schreibdurchsatz in Rechnung gestellt wird und wie Sie die Kapazität verwalten.

Für On-Demand-Modustabellen müssen Sie nicht angeben, wie viel Lese- und Schreibdurchsatz Sie von Ihrer Anwendung erwarten. Amazon Keyspaces berechnet Ihnen die Lese- und Schreibvorgänge, die Ihre Anwendung an Ihren Tabellen durchführt, in Form von Leseanforderungseinheiten und Schreibanforderungseinheiten. Wenn auf Ihrer Tabelle keine Aktivität stattfindet, zahlen Sie nicht für den Durchsatz, aber es fallen trotzdem Speichergebühren an.

## Löschen von Tabellen

Wenn Sie eine unbenutzte Tabelle entdeckt haben und diese löschen möchten, sollten Sie zunächst eine Sicherungskopie erstellen oder die Daten exportieren.

Durchgeführte Backups AWS Backup können Cold Storage Tiering nutzen und so die Kosten weiter senken. In der Dokumentation zur [Verwaltung von Backup-Plänen](#) finden Sie Informationen dazu, wie Sie einen Lebenszyklus verwenden können, um Ihr Backup in einen Cold Storage zu verschieben.

Nachdem Ihre Tabelle gesichert wurde, können Sie sie entweder über die AWS Management Console oder über die AWS Command Line Interface löschen.

## Bereinigen ungenutzter point-in-time Wiederherstellungs-Backups (PITR)

Amazon Keyspaces bietet Point-in-time Recovery, das kontinuierliche Backups für 35 Tage bereitstellt, um Sie vor versehentlichen Schreib- oder Löschvorgängen zu schützen. Mit PITR-Backups sind Kosten verbunden.

In der Dokumentation unter finden Sie Informationen [point-in-time P-Wiederherstellung](#) darüber, ob für Ihre Tabellen Backups aktiviert sind, die möglicherweise nicht mehr benötigt werden.

## Auswerten Ihrer Tabellennutzungsmuster

Dieser Abschnitt bietet einen Überblick darüber, wie Sie beurteilen können, ob Sie Ihre Amazon Keyspaces-Tabellen effizient verwenden. Es gibt bestimmte Nutzungsmuster, die für Amazon Keyspaces nicht optimal sind, und sie bieten Raum für Optimierungen sowohl in Bezug auf die Leistung als auch in Bezug auf die Kosten.

### Themen

- [Ausführen von weniger strikt konsistenten Lesevorgängen](#)
- [Aktivieren von Time to Live \(TTL\)](#)

## Ausführen von weniger strikt konsistenten Lesevorgängen

Amazon Keyspaces ermöglicht es Ihnen, die [Lesekonsistenz](#) pro Anfrage zu konfigurieren. Leseanforderungen sind standardmäßig letztendlich konsistent. Eventuell werden konsistente Lesevorgänge mit 0,5 RCU für bis zu 4 KB Daten berechnet.

Die meisten Teile von verteilten Workloads sind flexibel und können letztendliche Konsistenz tolerieren. Es kann jedoch Zugriffsmuster geben, die strikt konsistente Lesevorgänge erfordern. Für stark konsistente Lesevorgänge werden 1 RCU für bis zu 4 KB Daten berechnet, wodurch sich Ihre

Lesekosten im Wesentlichen verdoppeln. Amazon Keyspaces bietet Ihnen die Flexibilität, beide Konsistenzmodelle in derselben Tabelle zu verwenden.

Durch Auswertung Ihres Workloads und Anwendungscodes können Sie prüfen, ob strikt konsistente Lesevorgänge nur bei Bedarf verwendet werden.

## Aktivieren von Time to Live (TTL)

[Time to Live \(TTL\)](#) hilft Ihnen dabei, Ihre Anwendungslogik zu vereinfachen und den Speicherpreis zu optimieren, indem Daten aus Tabellen automatisch ablaufen. Daten, die Sie nicht mehr benötigen, werden basierend auf dem von Ihnen festgelegten Time-to-Live-Wert automatisch aus Ihrer Tabelle gelöscht.

## Auswerten Ihrer bereitgestellten Kapazität, um eine angemessene Bereitstellung zu erzielen

Dieser Abschnitt bietet einen Überblick darüber, wie Sie beurteilen können, ob Ihre Amazon Keyspaces-Tabellen über die richtige Größe für die Bereitstellung verfügen. Wenn sich Ihre Arbeitslast weiterentwickelt, sollten Sie Ihre Betriebsabläufe entsprechend ändern, insbesondere wenn Ihre Amazon Keyspaces-Tabelle im Bereitstellungsmodus konfiguriert ist und Sie das Risiko einer Über- oder Unterbereitstellung Ihrer Tabellen eingehen.

Die in diesem Abschnitt beschriebenen Verfahren erfordern statistische Informationen, die aus den Amazon Keyspaces-Tabellen erfasst werden sollten, die Ihre Produktionsanwendung unterstützen. Um Ihr Anwendungsverhalten zu verstehen, sollten Sie einen Zeitraum definieren, der signifikant genug ist, um die saisonale Datenabhängigkeit Ihrer Anwendung zu erfassen. Wenn Ihre Anwendung beispielsweise wöchentliche Muster aufweist, sollte ein Zeitraum von drei Wochen ausreichen, um die Anforderungen an den Anwendungsdurchsatz zu analysieren.

Wenn Sie nicht wissen, wo Sie anfangen sollen, verwenden Sie für die folgenden Berechnungen die Datennutzung von mindestens einem Monat.

Bei der Bewertung der Kapazität können Sie für Amazon Keyspaces-Tabellen Read Capacity Units (RCUs) und Write Capacity Units (WCU) unabhängig voneinander konfigurieren.

### Themen

- [So rufen Sie Verbrauchsmetriken aus Ihren Amazon Keyspaces-Tabellen ab](#)
- [So identifizieren Sie Amazon Keyspaces-Tabellen mit unzureichender Bereitstellung](#)
- [So identifizieren Sie übermäßig bereitgestellte Amazon Keyspaces-Tabellen](#)

## So rufen Sie Verbrauchsmetriken aus Ihren Amazon Keyspaces-Tabellen ab

Um die Tabellenkapazität zu bewerten, überwachen Sie die folgenden CloudWatch Kennzahlen und wählen Sie die entsprechende Dimension aus, um Tabelleninformationen abzurufen:

|                              |                               |
|------------------------------|-------------------------------|
| Lesekapazitätseinheiten      | Schreibkapazitätseinheiten    |
| ConsumedReadCapacityUnits    | ConsumedWriteCapacityUnits    |
| ProvisionedReadCapacityUnits | ProvisionedWriteCapacityUnits |
| ReadThrottleEvents           | WriteThrottleEvents           |

Sie können dies entweder über die AWS CLI oder die tun AWS Management Console.

### AWS CLI

Bevor Sie die Kennzahlen zum Tabellenverbrauch abrufen, müssen Sie zunächst einige historische Datenpunkte mithilfe der CloudWatch API erfassen.

Erstellen Sie zunächst zwei Dateien: `write-calc.json` und `read-calc.json`. Diese Dateien stellen die Berechnungen für die Tabelle dar. Sie müssen einige der Felder aktualisieren, wie in der Tabelle unten angegeben, damit sie zu Ihrer Umgebung passen.

#### Note

Wenn der Tabellename in Ihrem Konto nicht eindeutig ist, müssen Sie auch den Namen des Schlüsselraums angeben.

| Feldname                        | Definition                                            | Beispiel                                                           |
|---------------------------------|-------------------------------------------------------|--------------------------------------------------------------------|
| <code>&lt;table-name&gt;</code> | Der Name der Tabelle, die Sie analysieren             | SampleTable                                                        |
| <code>&lt;period&gt;</code>     | Der Zeitraum, den Sie zur Bewertung des Nutzungsziels | Für einen Zeitraum von 1 Stunde müssen Sie Folgendes angeben: 3600 |



| Feldname     | Definition                                                          | Beispiel            |
|--------------|---------------------------------------------------------------------|---------------------|
|              | verwenden, basierend auf Sekunden                                   |                     |
| <start-time> | Der Beginn Ihres Auswertungsintervalls, im ISO8601-Format angegeben | 2022-02-21T23:00:00 |
| <end-time>   | Das Ende Ihres Auswertungsintervalls, im ISO8601-Format angegeben   | 2022-02-22T06:00:00 |

Die Datei mit Schreibberechnungen ruft die Anzahl der WCU ab, die im angegebenen Zeitraum für den angegebenen Zeitraum bereitgestellt und verbraucht wurden. Außerdem wird ein Nutzungsprozentsatz generiert, der für Analysen verwendet werden kann. Der vollständige Inhalt der `write-calc.json` Datei sollte wie im folgenden Beispiel aussehen.

```
{
 "MetricDataQueries": [
 {
 "Id": "provisionedWCU",
 "MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ProvisionedWriteCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Average"
 },
 "Label": "Provisioned",
 "ReturnData": false
 },
 {
 "Id": "consumedWCU",
```

```

"MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ConsumedWriteCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Sum"
},
"Label": "",
"ReturnData": false
},
{
 "Id": "m1",
 "Expression": "consumedWCU/PERIOD(consumedWCU)",
 "Label": "Consumed WCUs",
 "ReturnData": false
},
{
 "Id": "utilizationPercentage",
 "Expression": "100*(m1/provisionedWCU)",
 "Label": "Utilization Percentage",
 "ReturnData": true
}
],
"StartTime": "<start-time>",
"EndTime": "<end-time>",
"ScanBy": "TimestampDescending",
"MaxDatapoints": 24
}

```

Die gelesene Berechnungsdatei verwendet ähnliche Metriken. Diese Datei ruft ab, wie viele RCUs während des angegebenen Zeitraums bereitgestellt und genutzt wurden. Der Inhalt der `read-calc.json` Datei sollte wie in diesem Beispiel aussehen.

```

{
 "MetricDataQueries": [
 {

```

```

 "Id": "provisionedRCU",
 "MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ProvisionedReadCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Average"
 },
 "Label": "Provisioned",
 "ReturnData": false
 },
 {
 "Id": "consumedRCU",
 "MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ConsumedReadCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Sum"
 },
 "Label": "",
 "ReturnData": false
 },
 {
 "Id": "m1",
 "Expression": "consumedRCU/PERIOD(consumedRCU)",
 "Label": "Consumed RCUs",
 "ReturnData": false
 },
 {

```

```

 "Id": "utilizationPercentage",
 "Expression": "100*(m1/provisionedRCU)",
 "Label": "Utilization Percentage",
 "ReturnData": true
 }
],
"StartTime": "<start-time>",
"EndTime": "<end-time>",
"ScanBy": "TimestampDescending",
"MaxDatapoints": 24
}

```

Sobald Sie die Dateien erstellt haben, können Sie mit dem Abrufen der Nutzungsdaten beginnen.

1. Geben Sie den folgenden Befehl ein, um die Schreibauslastungsdaten abzurufen:

```
aws cloudwatch get-metric-data --cli-input-json file://write-calc.json
```

2. Geben Sie den folgenden Befehl ein, um die Lesenutzungsdaten abzurufen:

```
aws cloudwatch get-metric-data --cli-input-json file://read-calc.json
```

Das Ergebnis für beide Abfragen ist eine Reihe von Datenpunkten im JSON-Format, die für Analysen verwendet werden können. Ihre Ergebnisse hängen von der Anzahl der von Ihnen angegebenen Datenpunkte, dem Zeitraum und Ihren eigenen spezifischen Workload-Daten ab. Es könnte wie im folgenden Beispiel aussehen.

```

{
 "MetricDataResults": [
 {
 "Id": "utilizationPercentage",
 "Label": "Utilization Percentage",
 "Timestamps": [
 "2022-02-22T05:00:00+00:00",
 "2022-02-22T04:00:00+00:00",
 "2022-02-22T03:00:00+00:00",
 "2022-02-22T02:00:00+00:00",
 "2022-02-22T01:00:00+00:00",
 "2022-02-22T00:00:00+00:00",
 "2022-02-21T23:00:00+00:00"
]
 },

```

```
 "Values": [
 91.55364583333333,
 55.066631944444445,
 2.6114930555555556,
 24.9496875,
 40.947256944444445,
 25.618194444444444,
 0.0
],
 "StatusCode": "Complete"
 }
],
"Messages": []
}
```

### Note

Wenn Sie einen kurzen Zeitraum und einen langen Zeitraum angeben, müssen Sie möglicherweise den MaxDatapoints Wert ändern, der im Skript standardmäßig auf 24 gesetzt ist. Dies entspricht einem Datenpunkt pro Stunde und 24 pro Tag.

## AWS Management Console

1. Melden Sie sich bei an AWS Management Console und navigieren Sie zur CloudWatch Serviceseite unter [Erste Schritte mit dem AWS Management Console](#). Wählen Sie bei AWS-Region Bedarf das entsprechende aus.
2. Suchen Sie in der linken Navigationsleiste den Abschnitt Metriken und wählen Sie Alle Metriken aus.
3. Dadurch wird ein Dashboard mit zwei Bereichen geöffnet. Im oberen Bereich wird die Grafik angezeigt, und im unteren Bereich befinden sich die Kennzahlen, die Sie grafisch darstellen möchten. Wählen Sie das Amazon Keyspaces-Panel.
4. Wählen Sie in den Unterfenstern die Kategorie „Tabellenmetriken“ aus. Dies zeigt Ihnen die Tabellen in Ihrer aktuellen Version AWS-Region.
5. Ermitteln Sie Ihren Tabellennamen, indem Sie im Menü nach unten scrollen und die Metriken für Schreibvorgänge auswählen: ConsumedWriteCapacityUnits und ProvisionedWriteCapacityUnits.

**Note**

In diesem Beispiel geht es um Metriken für Schreibvorgänge, Sie können diese Schritte jedoch auch verwenden, um die Metriken für Lesevorgänge grafisch darzustellen.

- Wählen Sie die Registerkarte Graphed metrics (2) (Grafisch dargestellte Metriken (2)) aus, um die Formeln zu ändern. CloudWatch Wählt standardmäßig die Statistikfunktion Durchschnitt für die Grafiken.
- Wenn beide grafisch dargestellten Metriken ausgewählt sind (Kontrollkästchen auf der linken Seite), wählen Sie das Menü Add math (Math. hinzufügen), dann Common (Allgemein) und schließlich die Funktion Percentage (Prozent) aus. Wiederholen Sie den Vorgang zweimal.

Wählen Sie zum ersten Mal die Prozentfunktion aus.

Wählen Sie zum zweiten Mal die Funktion Percentage aus.

- Zu diesem Zeitpunkt sollten vier Metriken im unteren Menü aufgeführt sein. Befassen wir uns nun mit der Berechnung von ConsumedWriteCapacityUnits. Um konsistent zu sein, müssen Sie die Namen mit denen abgleichen, die Sie in dem AWS CLI Abschnitt verwendet haben. Klicken Sie auf die m1 ID und ändern Sie diesen Wert in consumedWCU.
- Ändern Sie die Statistik von Average (Durchschnitt) in Sum (Summe). Diese Aktion erstellt automatisch eine weitere Metrik namens ANOMALY\_DETECTION\_BAND. Im Rahmen dieses Verfahrens können Sie dies ignorieren, indem Sie das Kontrollkästchen für die neu generierte ad1-Metrik entfernen.
- Wiederholen Sie Schritt 8, um die m2 ID in provisionedWCU umzubenennen. Lassen Sie die Statistik auf Average (Durchschnitt) eingestellt.
- Wählen Sie das Label Expression1 und aktualisieren Sie den Wert auf m1 und das Label auf Consumed WCUs.

**Note**

Stellen Sie sicher, dass Sie nur m1 (Kontrollkästchen links) und provisionedWCU ausgewählt haben, damit die Daten richtig angezeigt werden. Aktualisieren Sie die Formel, indem Sie auf Details klicken und die Formel in `consumedWCU/PERIOD(consumedWCU)` ändern. Durch diesen Schritt wird möglicherweise auch

eine weitere ANOMALY\_DETECTION\_BAND-Metrik generiert, die Sie jedoch im Rahmen dieses Verfahrens ignorieren können.

12. Sie sollten jetzt über zwei Grafiken verfügen: eine, die Ihre bereitgestellten WCUs in der Tabelle anzeigt, und eine andere, die die verbrauchten WCUs anzeigt.
13. Aktualisieren Sie die Prozentformel, indem Sie die Grafik für Expression2 (e2) auswählen. Benennen Sie die Labels und IDs in utilizationPercentage um. Benennen Sie die Formel so um, dass sie  $100 * (m1 / \text{provisionedWCU})$  entspricht.
14. Entfernen Sie das Kontrollkästchen aus allen Metriken außer UtilizationPercentage, um Ihre Nutzungsmuster zu visualisieren. Das Standardintervall ist auf 1 Minute festgelegt, Sie können es jedoch nach Bedarf ändern.

Die Ergebnisse, die Sie erhalten, hängen von den tatsächlichen Daten Ihres Workloads ab. Intervalle mit einer Auslastung von mehr als 100% sind anfällig für Fehler bei geringer Durchsatzkapazität. Amazon Keyspaces bietet [Burst-Kapazität](#), aber sobald die Burst-Kapazität erschöpft ist, kommt es bei einem Wert über 100% zu Fehlern bei niedriger Durchsatzkapazität.

## So identifizieren Sie Amazon Keyspaces-Tabellen mit unzureichender Bereitstellung

Bei den meisten Workloads gilt eine Tabelle als unzureichend bereitgestellt, wenn sie ständig mehr als 80% der bereitgestellten Kapazität beansprucht.

[Burst Capacity](#) ist eine Funktion von Amazon Keyspaces, die es Kunden ermöglicht, vorübergehend mehr RCUs/WCUs als ursprünglich bereitgestellt zu verbrauchen (mehr als den bereitgestellten Durchsatz pro Sekunde, der für die Tabelle definiert wurde). Die Burst-Kapazität soll plötzliche Zunahmen des Datenverkehrs aufgrund von besonderen Ereignissen oder Auslastungsspitzen auffangen. Diese Burst-Kapazität ist begrenzt. Weitere Informationen finden Sie unter [the section called "Burst-Kapazität"](#). Sobald die ungenutzten RCUs und WCUs aufgebraucht sind, kann es zu Fehlern beim Durchsatz mit niedriger Kapazität kommen, wenn Sie versuchen, mehr Kapazität als vorgesehen zu verbrauchen. Wenn sich Ihr Anwendungsdatenverkehr der Auslastungsrate von 80% nähert, ist das Risiko, dass Fehler bei geringem Kapazitätsdurchsatz auftreten, erheblich höher.

Die Regel der Auslastungsrate von 80 % hängt von der Saisonalität Ihrer Daten und dem Wachstum Ihres Datenverkehrs ab. Betrachten Sie folgende Szenarien:

- Wenn Ihr Datenverkehr in den letzten 12 Monaten bei einer Auslastung von ~ 90 % stabil war, hat Ihre Tabelle genau die richtige Kapazität.

- Wenn Ihr Anwendungsdatenverkehr monatlich um 8 % wächst, werden Sie in weniger als 3 Monaten 100 % erreichen.
- Auch wenn Ihr Anwendungsdatenverkehr monatlich um 5 % wächst, werden Sie in etwas mehr als 4 Monaten 100 % erreichen.

Die Ergebnisse der obigen Abfragen vermitteln ein Bild Ihrer Auslastungsrate. Verwenden Sie sie als Orientierungshilfe für die Auswertung weiterer Metriken, die bei der Entscheidung, Ihre Tabellenkapazität nach Bedarf zu erhöhen, hilfreich sein können (z. B. monatliche oder wöchentliche Wachstumsrate). Legen Sie gemeinsam mit Ihrem Operations-Team fest, welcher Prozentsatz für Ihren Workload und Ihre Tabellen geeignet ist.

Es gibt spezielle Szenarien, in denen die Daten verzerrt sind, wenn Sie sie täglich oder wöchentlich analysieren. Beispielsweise könnten Sie bei saisonalen Anwendungen, die während der Arbeitszeit stark ausgelastet sind (aber außerhalb der Geschäftszeiten auf nahezu Null sinken), von der [Planung der auto-scaling der Anwendung](#) profitieren, bei der Sie die Tageszeiten (und Wochentage) angeben, um die bereitgestellte Kapazität zu erhöhen, sowie den Zeitpunkt, zu dem sie reduziert werden soll. Anstatt eine höhere Kapazität anzustreben, um die geschäftigen Zeiten abzudecken, können Sie auch von den [auto-scaling Skalierungskonfigurationen der Amazon Keyspaces-Tabellen](#) profitieren, wenn Ihre Saisonalität weniger ausgeprägt ist.

## So identifizieren Sie übermäßig bereitgestellte Amazon Keyspaces-Tabellen

Die mit den obigen Skripten erhaltenen Abfrageergebnisse liefern die für eine erste Analyse erforderlichen Datenpunkte. Wenn Ihr Datensatz für mehrere Intervalle Auslastungswerte von weniger als 20 % aufweist, wurde für Ihre Tabelle möglicherweise zu viel Kapazität bereitgestellt. Um genauer herauszufinden, ob Sie die Anzahl der WCUs und RCUs reduzieren müssen, sollten Sie die anderen Messwerte in den Intervallen erneut überprüfen.

Wenn Ihre Tabelle mehrere niedrige Nutzungsintervalle enthält, können Sie von der Verwendung von Application Auto Scaling-Richtlinien profitieren, indem Sie entweder Application Auto Scaling planen oder einfach die standardmäßigen Application Auto Scaling Scaling-Richtlinien für die Tabelle konfigurieren, die auf der Auslastung basieren.

Wenn Sie eine Arbeitslast mit einem niedrigen Verhältnis zwischen Auslastung und hoher Drosselung haben (Max (ThrottleEventsThrottleEvents) /Min () im Intervall), kann dies passieren, wenn Sie eine sehr hohe Arbeitslast haben, bei der der Verkehr an bestimmten Tagen (oder Tageszeiten) erheblich zunimmt, ansonsten aber konstant niedrig ist. In diesen Szenarien kann es von Vorteil sein, [geplantes Application Auto Scaling](#) zu verwenden.



# Verwenden von NoSQL Workbench mit Amazon Keyspaces (für Apache Cassandra)

NoSQL Workbench ist eine clientseitige Anwendung, mit der Sie nichtrelationale Datenmodelle für Amazon Keyspaces einfacher entwerfen und visualisieren können. NoSQL Workbench Clients sind für Windows, macOS und Linux verfügbar.

## Automatisches Entwerfen von Datenmodellen und Erstellen von Ressourcen

NoSQL Workbench bietet Ihnen eine point-and-click Schnittstelle zum Entwerfen und Erstellen von Amazon Keyspaces-Datenmodellen. Sie können ganz einfach neue Datenmodelle von Grund auf neu erstellen, indem Sie Schlüsselräume, Tabellen und Spalten definieren. Sie können auch vorhandene Datenmodelle importieren und Änderungen vornehmen (z. B. Spalten hinzufügen, bearbeiten oder entfernen), um die Datenmodelle für neue Anwendungen anzupassen. NoSQL Workbench ermöglicht es Ihnen dann, die Datenmodelle an Amazon Keyspaces oder Apache Cassandra zu übergeben und die Schlüsselräume und Tabellen automatisch zu erstellen. Informationen zum Erstellen von Datenmodellen finden Sie unter [the section called “Data Modeler”](#).

## Visualisieren von Datenmodellen

Mit NoSQL Workbench können Sie Ihre Datenmodelle visualisieren, um sicherzustellen, dass die Datenmodelle die Abfragen und Zugriffsmuster Ihrer Anwendung unterstützen. Sie können Ihre Datenmodelle auch in einer Vielzahl von Formaten für Zusammenarbeit, Dokumentation und Präsentationen speichern und exportieren. Weitere Informationen finden Sie unter [the section called “Data Visualizer”](#).

## Themen

- [NoSQL Workbench herunterladen](#)
- [Erste Schritte mit NoSQL Workbench](#)
- [So erstellen Sie Datenmodelle](#)
- [So visualisieren Sie Datenmodelle](#)
- [So übertragen Sie Datenmodelle an Amazon Keyspaces und Apache Cassandra](#)
- [Beispieldatenmodelle in NoSQL Workbench](#)
- [Versionsverlauf für NoSQL-Workbench](#)

# NoSQL Workbench herunterladen

Befolgen Sie diese Anweisungen zum Herunterladen und Installieren von NoSQL Workbench.

## Herunterladen und Installieren von NoSQL Workbench

1. Verwenden Sie einen der folgenden Links zum kostenlosen Herunterladen von NoSQL Workbench.

| Betriebssystem | Download-Link                             |
|----------------|-------------------------------------------|
| macOS          | <a href="#">Für macOS herunterladen</a>   |
| Linux*         | <a href="#">Download für Linux</a>        |
| Windows        | <a href="#">Für Windows herunterladen</a> |

\* NoSQL-Workbench unterstützt Ubuntu 12.04, Fedora 21 und Debian 8 sowie sämtliche neueren Versionen dieser Linux-Verteilungen.

2. Starten Sie nach Abschluss des Downloads die Anwendung und folgen Sie den Anweisungen auf dem Bildschirm, um die Installation abzuschließen.

## Erste Schritte mit NoSQL Workbench

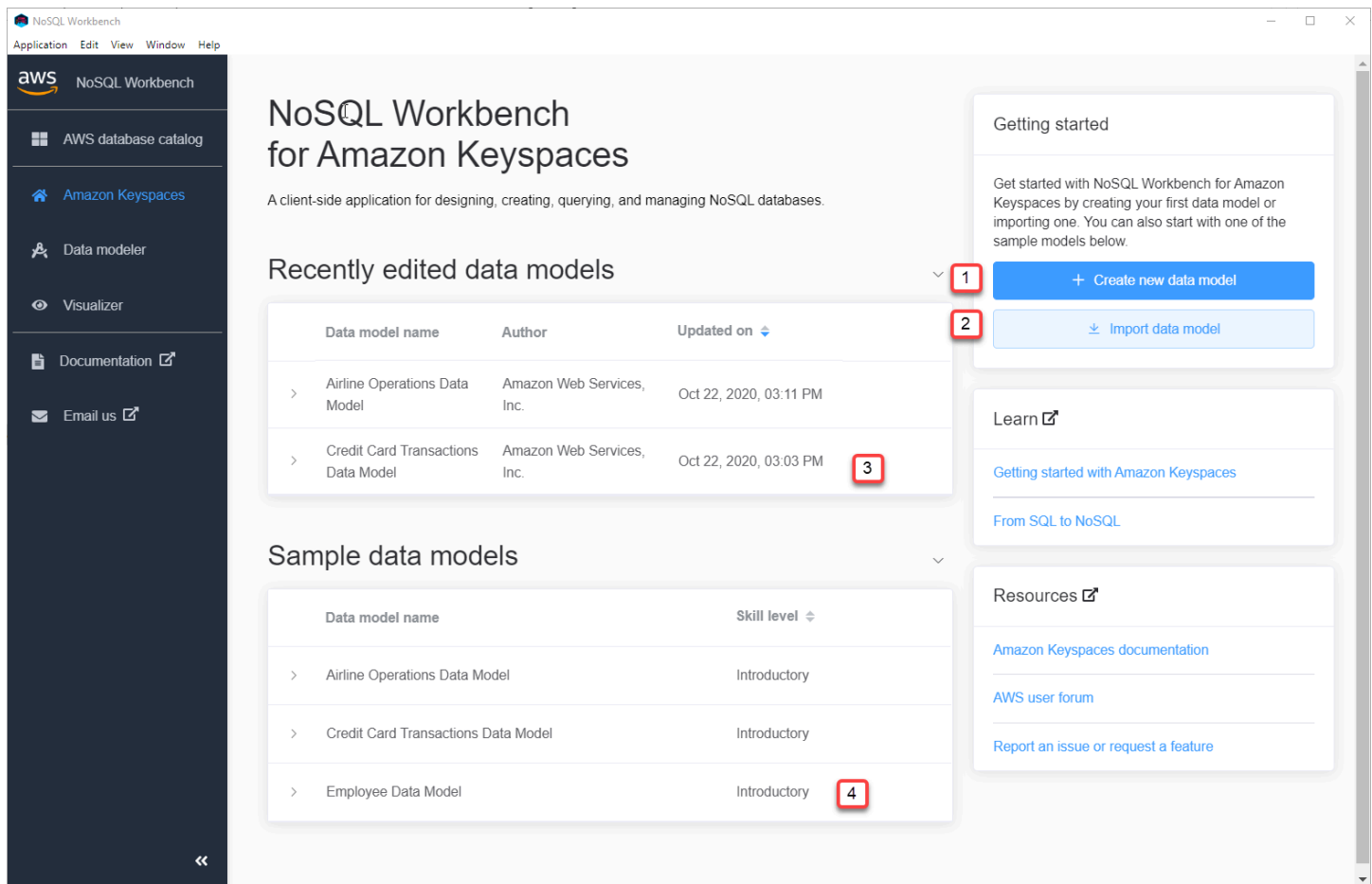
Um mit NoSQL Workbench zu beginnen, wählen Sie auf der Seite Datenbankkatalog in NoSQL Workbench Amazon Keyspaces und dann Launch aus.

The screenshot shows the AWS NoSQL Workbench application window. At the top, it says "aws NoSQL Workbench" and "A client-side application for designing, creating, querying, and managing NoSQL databases." Below this, there's a "How it works" section with three main features: "Data modeler" (Build new data models, Add tables and indexes, Import and export models), "Visualizer" (Add sample data, Visualize data layout and structure, Commit model the cloud), and "Operation builder" (Build operations and queries, Use a guided form, Generate code for data-plane operations). The "Get started by choosing a database service" section has two options: "Amazon DynamoDB" and "Amazon Keyspaces (for Apache Cassandra)". The latter is highlighted with a red box. Below each option is a table with details like Type, Description, and Use cases. At the bottom, there's a small legal disclaimer.

By using NoSQL Workbench you agree to the [License Agreement](#), [AWS Customer Agreement](#), [AWS Service Terms](#), [AWS Privacy Notice](#), and [Source Code Notice](#). If you already have an AWS Customer Agreement, you agree that the terms of that agreement govern your installation and use of this product. See also [third party notices](#).

Dadurch wird die NoSQL Workbench-Startseite für Amazon Keyspaces geöffnet, auf der Sie die folgenden Optionen haben, um loszulegen:

1. Erstellen Sie ein neues Datenmodell.
2. Importiert ein vorhandenes Datenmodell im JSON-Format.
3. Öffnen Sie ein kürzlich bearbeitetes Datenmodell.
4. Öffnen Sie eines der verfügbaren Beispielm Modelle.



Jede der Optionen öffnet den NoSQL Workbench-Datenmodellierer. Informationen zum weiteren Erstellen eines neuen Datenmodells finden Sie unter [the section called “Erstellen eines Datenmodells”](#). Informationen zum Bearbeiten eines vorhandenen Datenmodells finden Sie unter [the section called “Bearbeiten eines Datenmodells”](#).

## So erstellen Sie Datenmodelle

Sie können den NoSQL Workbench-Datenmodellierer verwenden, um neue Datenmodelle zu entwerfen, die auf den Datenzugriffsmustern Ihrer Anwendung basieren. Sie können den Datenmodellierer verwenden, um neue Datenmodelle zu entwerfen oder vorhandene Datenmodelle zu importieren und zu ändern, die mit NoSQL Workbench erstellt wurden. Der Datenmodellierer enthält auch einige Beispieldatenmodelle, die Ihnen den Einstieg in die Datenmodellierung erleichtern.

### Themen

- [Erstellen neuer Datenmodelle mit NoSQL Workbench](#)

- [Bearbeiten vorhandener Datenmodelle mit NoSQL Workbench](#)

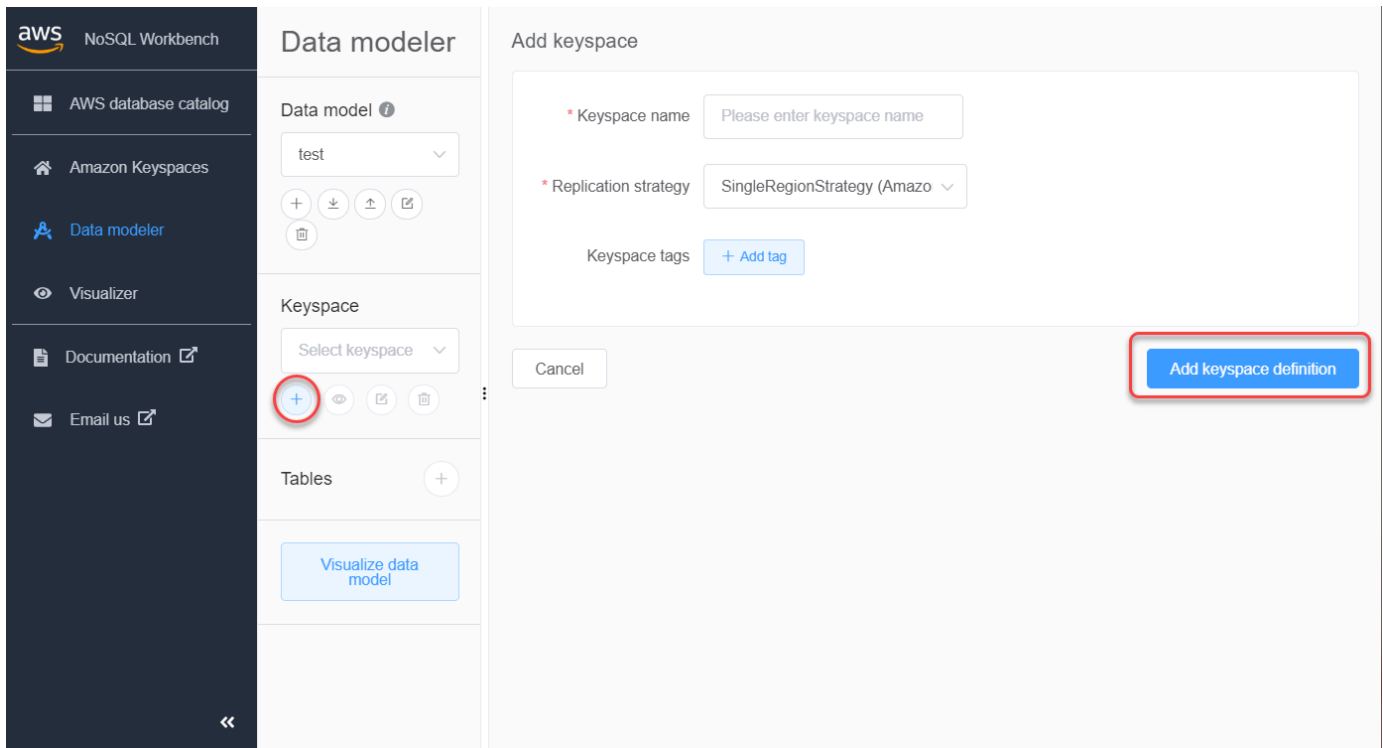
## Erstellen neuer Datenmodelle mit NoSQL Workbench

Um ein neues Datenmodell für Amazon Keyspaces zu erstellen, können Sie den NoSQL Workbench-Datenmodeller verwenden, um Schlüsselräume, Tabellen und Spalten zu erstellen. Führen Sie diese Schritte aus, um ein neues Datenmodell zu erstellen.

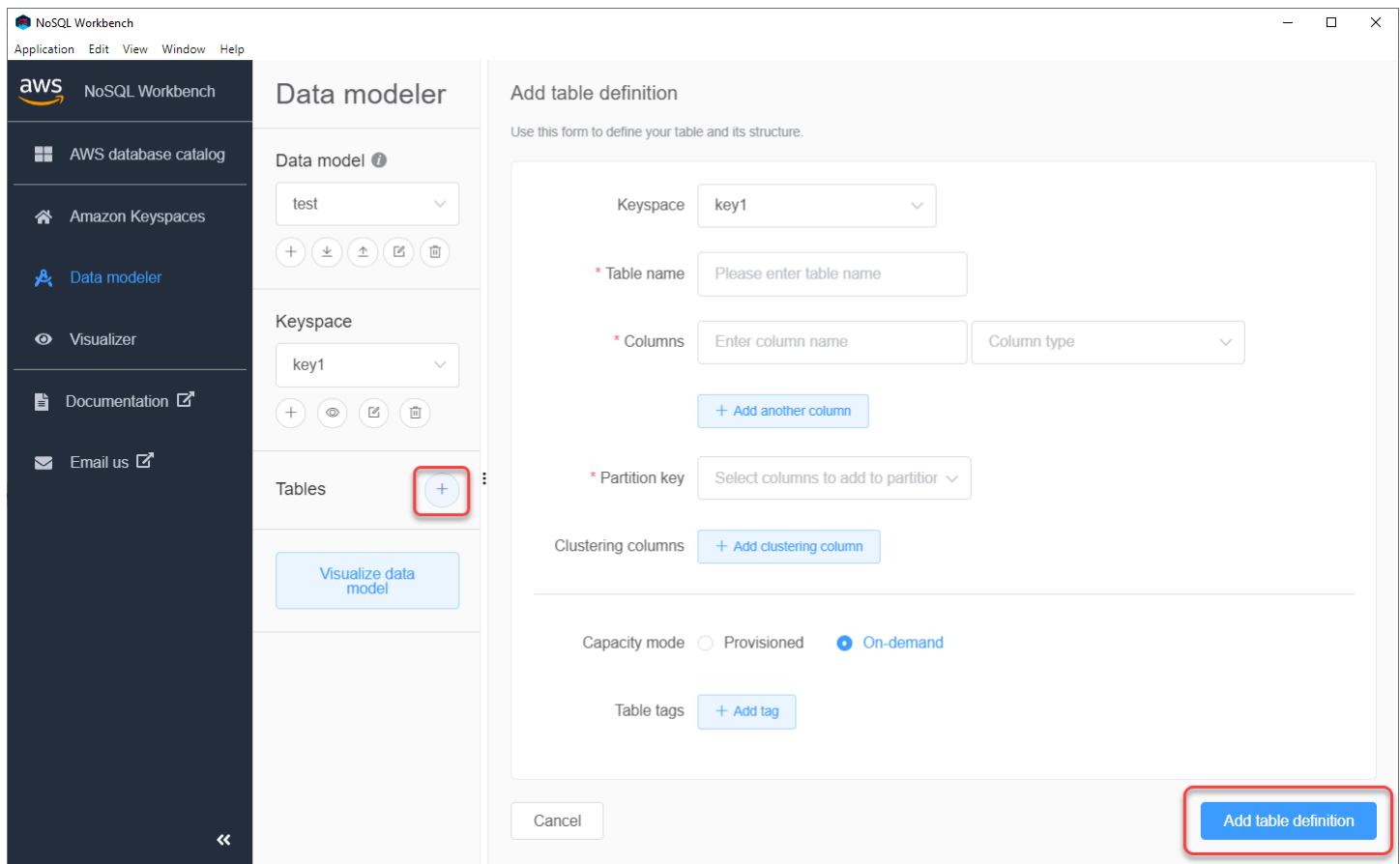
1. Um ein neues Keyspace zu erstellen, wählen Sie das Pluszeichen unter Keyspace.

Wählen Sie in diesem Schritt die folgenden Eigenschaften und Einstellungen aus.

- **Keyspace-Name** — Geben Sie den Namen des neuen Schlüsselraums ein.
  - **Replikationsstrategie** — Wählen Sie die Replikationsstrategie für den Keyspace. Amazon Keyspaces verwendet die `SingleRegionStrategy`, um Daten dreimal automatisch in mehreren AWS Availability Zones zu replizieren. Wenn Sie planen, das Datenmodell in einen Apache Cassandra-Cluster zu übertragen, können Sie wählen `SimpleStrategy` oder `NetworkTopologyStrategy`.
  - **Keyspaces-Tags** — Ressourcen sind optional und können Sie Ressourcen auf verschiedene Weise auf verschiedene Weise auf verschiedene Weise kategorisieren, z. B. nach Zweck, Eigentümer, Umgebung oder anderen Kriterien. Weitere Informationen zu Tags für Amazon Keyspaces-Ressourcen finden Sie unter [the section called “Arbeiten mit Tags”](#).
2. Wählen Sie Keyspace-Definition hinzufügen, um den Keyspace zu erstellen.



3. Um eine neue Tabelle zu erstellen, wählen Sie das Pluszeichen neben Tabellen. In diesem Schritt definieren Sie die folgenden Eigenschaften und Einstellungen.
  - Tabellenname — Der Name der neuen Tabelle.
  - Spalten — Fügen Sie einen Spaltennamen hinzu und wählen Sie den Datentyp aus. Wiederholen Sie diese Schritte für jede Spalte in Ihrem Schema.
  - Partitionsschlüssel — Wählen Sie Spalten für den Partitionsschlüssel aus.
  - Clusterspalten — Wählen Sie Clusterspalten aus (optional).
  - Kapazmodus — Wählen Sie den LesSchreimodus für die Tabelle aus. Sie können zwischen bereitgestellter Kapazität und On-Demand-Kapazität wählen. Weitere Informationen zu Kapazmodus finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).
  - Tabellenbeschriftungen — Ressourcen sind optional und können Sie Ressourcen auf verschiedene Weise auf verschiedene Weise auf verschiedene Weise kategorisieren, z. B. nach Zweck, Eigentümer, Umgebung oder anderen Kriterien. Weitere Informationen zu Tags für Amazon Keyspaces-Ressourcen finden Sie unter [the section called “Arbeiten mit Tags”](#).
4. Wählen Sie Tabellendefinition hinzufügen, um die neue Tabelle zu erstellen.
5. Wiederholen Sie diese Schritte, um weitere Tabellen zu erstellen.
6. Fahren Sie mit [the section called “Visualisieren eines Datenmodells”](#) der Visualisierung des Datenmodells fort, das Sie erstellt haben.



## Bearbeiten vorhandener Datenmodelle mit NoSQL Workbench

Mit dem NoSQL Workbench-Datenmodeller können Sie bestehende Datenmodelle in Amazon Keyspaces bearbeiten. Dies können Datenmodelle sein, die aus einer Datei importiert wurden, die bereitgestellten Beispieldatenmodelle oder Datenmodelle, die Sie zuvor erstellt haben.

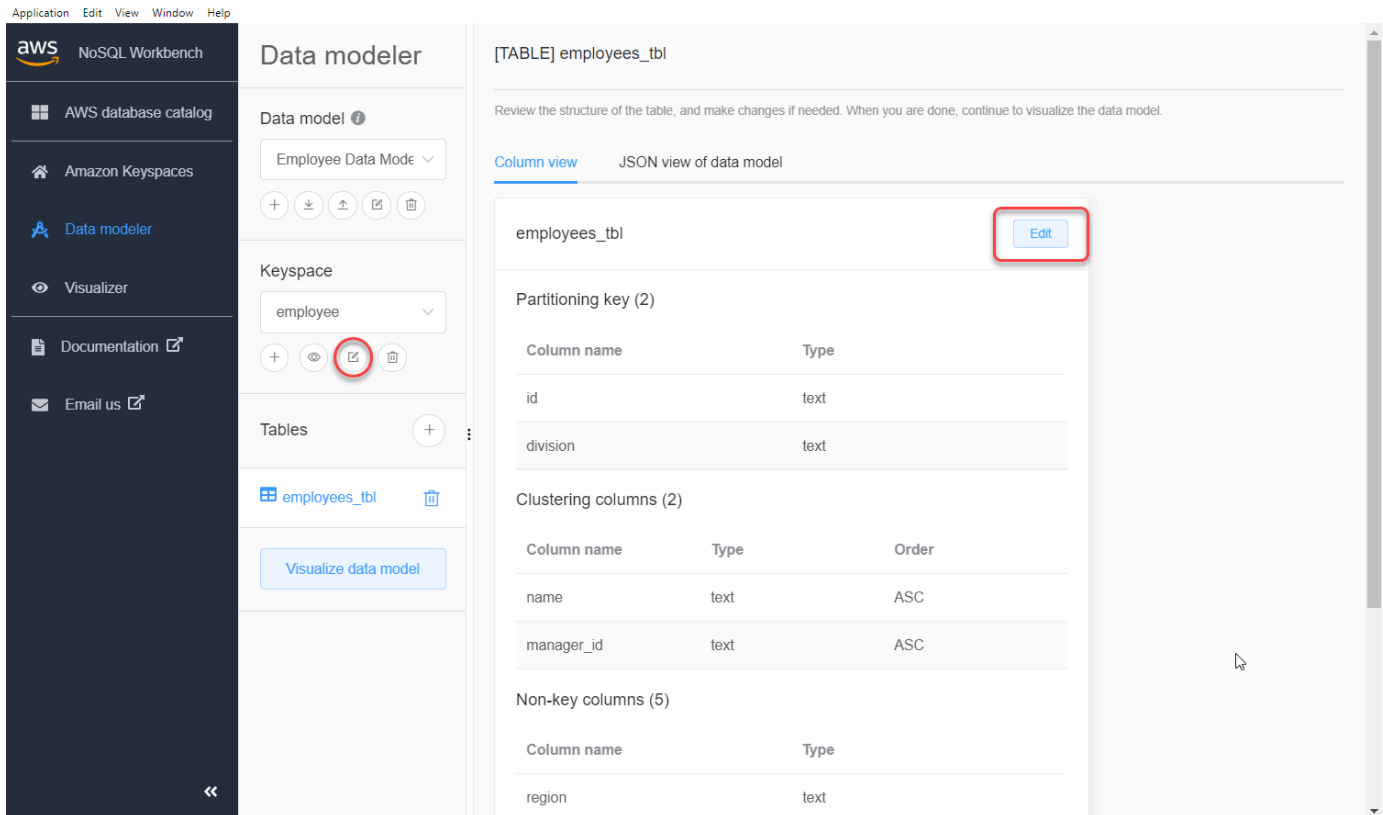
1. Um ein Keyspace zu bearbeiten, wählen Sie das Bearbeitungssymbol unter Keyspace.

In diesem Schritt können Sie die folgenden Eigenschaften und Einstellungen bearbeiten.

- Keyspace-Name — Geben Sie den Namen des neuen Schlüsselraums ein.
- Replikationsstrategie — Wählen Sie die Replikationsstrategie für den Keyspace. Amazon Keyspaces verwendet die SingleRegionStrategy, um Daten dreimal automatisch in mehreren AWS Availability Zones zu replizieren. Wenn Sie planen, das Datenmodell in einen Apache Cassandra-Cluster zu übertragen, können Sie wählen SimpleStrategy oder NetworkTopologyStrategy.
- Keyspaces-Tags — Ressourcen sind optional und können Sie Ressourcen auf verschiedene Weise auf verschiedene Weise auf verschiedene Weise kategorisieren, z. B. nach Zweck,

Eigentümer, Umgebung oder anderen Kriterien. Weitere Informationen zu Tags für Amazon Keyspaces-Ressourcen finden Sie unter [the section called “Arbeiten mit Tags”](#).

## 2. Wählen Sie Änderungen speichern, um den Keyspace zu aktualisieren.



## 3. Um eine Tabelle zu bearbeiten, wählen Sie neben dem Tabellennamen die Option Bearbeiten aus. In diesem Schritt können Sie die folgenden Eigenschaften und Einstellungen aktualisieren.

- Tabellennamen — Der Name der neuen Tabelle.
- Spalten — Fügen Sie einen Spaltennamen hinzu und wählen Sie den Datentyp aus. Wiederholen Sie diese Schritte für jede Spalte in Ihrem Schema.
- Partitionsschlüssel — Wählen Sie Spalten für den Partitionsschlüssel aus.
- Clusterspalten — Wählen Sie Clusterspalten aus (optional).
- Kapazmodus — Wählen Sie den LesSchreibmodus für die Tabelle aus. Sie können zwischen bereitgestellter Kapazität und On-Demand-Kapazität wählen. Weitere Informationen zu Kapazmodus finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).
- Tabellenbeschriftungen — Ressourcen sind optional und können Sie Ressourcen auf verschiedene Weise auf verschiedene Weise auf verschiedene Weise kategorisieren, z. B. nach Zweck, Eigentümer, Umgebung oder anderen Kriterien. Weitere Informationen zu Tags für Amazon Keyspaces-Ressourcen finden Sie unter [the section called “Arbeiten mit Tags”](#).



4. Wählen Sie Änderungen speichern, um die Tabelle zu aktualisieren.
5. Fahren Sie mit [the section called “Visualisieren eines Datenmodells”](#) der Visualisierung des Datenmodells fort, das Sie aktualisiert haben.

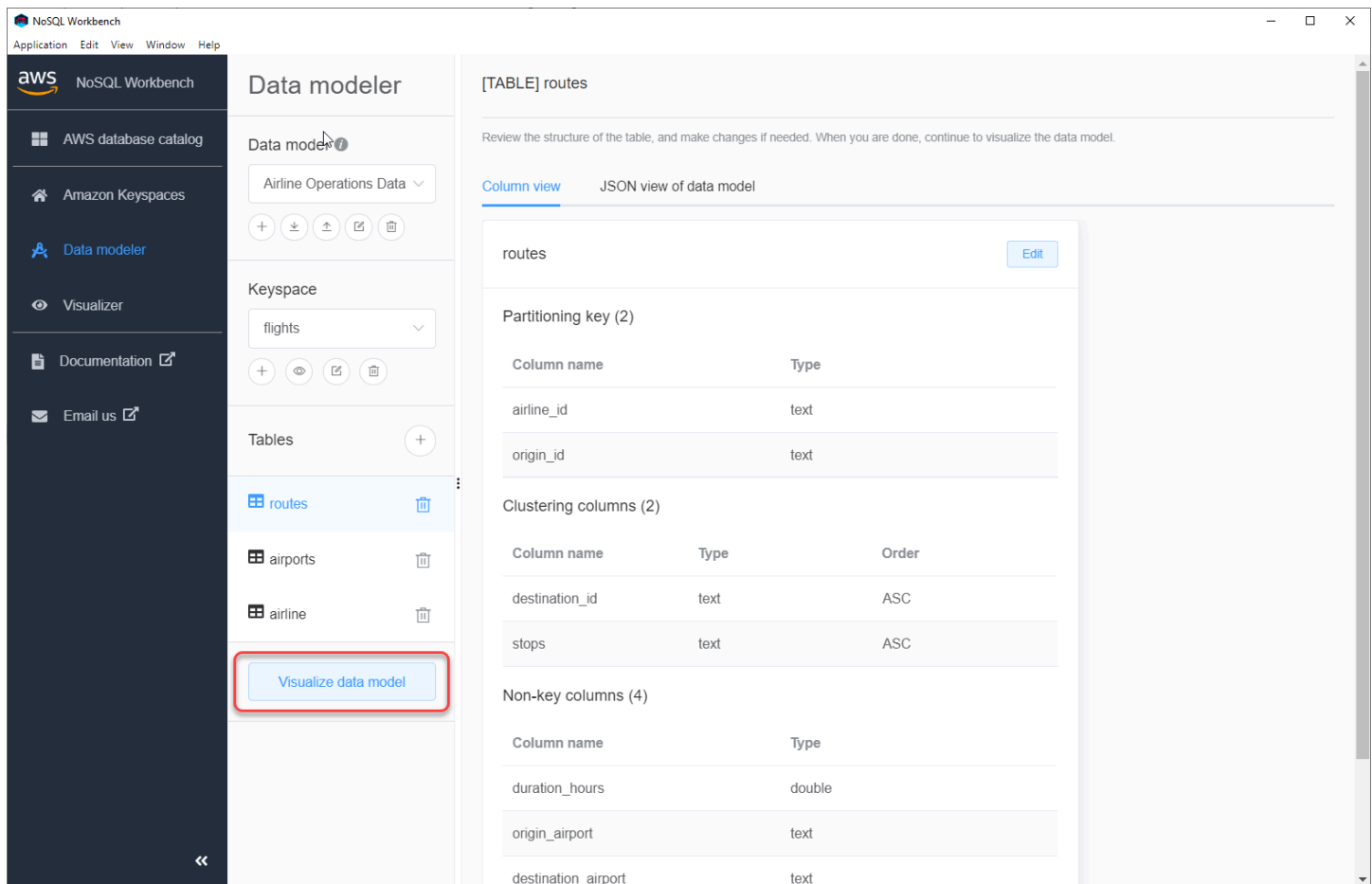
## So visualisieren Sie Datenmodelle

Mit NoSQL Workbench können Sie Ihre Datenmodelle visualisieren, um sicherzustellen, dass die Datenmodelle die Abfragen und Zugriffsmuster Ihrer Anwendung unterstützen. Sie können Ihre Datenmodelle auch in einer Vielzahl von Formaten für Zusammenarbeit, Dokumentation und Präsentationen speichern und exportieren.

Nachdem Sie ein neues Datenmodell erstellt oder ein vorhandenes Datenmodell bearbeitet haben, können Sie das Modell visualisieren.

### Visualisieren von Datenmodellen mit NoSQL Workbench

Wenn Sie das Datenmodell im Datenmodeler fertiggestellt haben, wählen Sie Datenmodell visualisieren.



The screenshot shows the AWS NoSQL Workbench interface. The left sidebar contains navigation options: 'Data modeler' (selected), 'Visualizer', 'Documentation', and 'Email us'. The main area displays the 'Data modeler' for the 'Airline Operations Data' keyspace, specifically for the 'flights' keyspace and the 'routes' table. The table structure is shown in 'Column view' with the following columns:

| Column name         | Type   |
|---------------------|--------|
| airline_id          | text   |
| origin_id           | text   |
| destination_id      | text   |
| stops               | text   |
| duration_hours      | double |
| origin_airport      | text   |
| destination_airport | text   |

The 'Partitioning key' section shows 'airline\_id' and 'origin\_id' as text columns. The 'Clustering columns' section shows 'destination\_id' and 'stops' as text columns, both ordered ASC. The 'Non-key columns' section shows 'duration\_hours' as a double column, and 'origin\_airport' and 'destination\_airport' as text columns. A red box highlights the 'Visualize data model' button in the sidebar.

Dadurch gelangen Sie zum Datenvisualisierer in NoSQL Workbench. Der Datenvisualisierer bietet eine visuelle Darstellung des Tabellenschemas und ermöglicht das Hinzufügen von Beispieldaten. Um einer Tabelle Beispieldaten hinzuzufügen, wählen Sie eine Tabelle aus dem Modell aus und klicken Sie dann auf Bearbeiten. Um eine neue Datenzeile hinzuzufügen, wählen Sie Neue Zeile hinzufügen am unteren Bildschirmrand aus. Klicken Sie auf Save (Speichern), wenn Sie fertig sind.

The screenshot shows the AWS NoSQL Workbench Visualizer interface. The left sidebar contains navigation options: AWS database catalog, Amazon Keyspaces, Data modeler, Visualizer (selected), Documentation, and Email us. The main area is titled 'Visualizer' and shows the 'Data model' for 'Airline Operations'. The 'Keyspace' is set to 'flights' and the 'Tables' list includes 'routes', 'airports', and 'airline'. The 'routes' table is selected, and its schema is displayed in a table format. The table has a primary key of 'airline\_id (text)' and clustering columns 'destination\_id (text)' and 'stops (text)'. Non-key columns are 'duration\_hours (double)' and 'origin\_airport (text)'. The table contains three rows of data. A '+ Add new row' button is highlighted with a red box at the bottom of the table.

| Primary key       |                         |                | Non-key columns         |                       |
|-------------------|-------------------------|----------------|-------------------------|-----------------------|
| Partition key     | Clustering columns      |                | duration_hours (double) | origin_airport (text) |
| airline_id (text) | destination_id (text) ↕ | stops (text) ↕ |                         |                       |
| acme_airlines     | MCI                     | 1              | 0.33333333              | Newark Liberty        |
| trusted_airlines  | MIC                     | 2              | 0.33333333              | Phoenix Sky Ha        |
| freedom_airline   | EWR                     | 1              | 0.33333333              | San Francisco         |

## Aggregierte Ansicht

Nachdem Sie das Schema der Tabelle bestätigt haben, können Sie Datenmodellvisualisierungen aggregieren.

The screenshot shows the AWS NoSQL Workbench Visualizer interface. The left sidebar contains navigation options: AWS database catalog, Amazon Keyspaces, Data modeler, Visualizer (selected), Documentation, and Email us. The main area is titled 'Visualizer' and shows the 'Data model' for 'Airline Operations Data' in the 'flights' Keyspace. The 'Tables' list includes 'routes', 'airports', and 'airline'. The 'Aggregate view' button is highlighted with a red box. The main table view shows the schema for the 'routes' table, which is a Primary key table. The schema is as follows:

| Primary key       |                  |                       |              | Non-key columns                  |                              |           |
|-------------------|------------------|-----------------------|--------------|----------------------------------|------------------------------|-----------|
| Partition key     |                  | Clustering columns    |              | Non-key columns                  |                              |           |
| airline_id (text) | origin_id (text) | destination_id (text) | stops (text) | origin_airport (text)            | destination_airport (text)   | equipment |
| acme_airlines     | EWR              | MCI                   | 1            | Newark Liberty International     | Kansas City International    | 737       |
| trusted_airlines  | PHX              | MIC                   | 2            | Phoenix Sky Harbor International | Kansas City International    | 737       |
| freedom_airlines  | SFO              | EWR                   | 1            | San Francisco International      | Newark Liberty International | 747       |

Nachdem Sie die Ansicht des Datenmodells aggregiert haben, können Sie die Ansicht in eine PNG-Datei exportieren. Um das Datenmodell in eine JSON-Datei zu exportieren, wählen Sie das Upload-Zeichen unter dem Namen des Datenmodells aus.

### Note

Sie können das Datenmodell jederzeit im Entwurfsprozess im JSON-Format exportieren.

The screenshot shows the NoSQL Workbench Visualizer interface. The main window displays the 'Aggregate view' of a table named '[TABLE] routes'. The table has a primary key consisting of 'airline\_id (text)', 'origin\_id (text)', 'destination\_id (text)', and 'stops (text)'. The data rows are as follows:

| Primary key       |                  |                       |              | Non-key columns                  |                              |         |
|-------------------|------------------|-----------------------|--------------|----------------------------------|------------------------------|---------|
| Partition key     |                  | Clustering columns    |              | Non-key columns                  |                              |         |
| airline_id (text) | origin_id (text) | destination_id (text) | stops (text) | origin_airport (text)            | destination_airport (text)   | equipme |
| acme_airlines     | EWR              | MCI                   | 1            | Newark Liberty International     | Kansas City International    | 737     |
| trusted_airlines  | PHX              | MIC                   | 2            | Phoenix Sky Harbor International | Kansas City International    | 737     |
| freedom_airlines  | SFO              | EWR                   | 1            | San Francisco International      | Newark Liberty International | 747     |

Below the table, there are two buttons: 'Commit to Amazon Keyspaces' and 'Commit to Apache Cassandra', both highlighted with red boxes. The 'Export to PNG' button in the top right corner is also highlighted with a red box.

Sie haben die folgenden Optionen, um die Änderungen zu übernehmen:

- Engagieren Sie sich für Amazon Keyspaces
- Entscheiden Sie sich für einen Apache Cassandra-Cluster

Weitere Informationen darüber, wie Sie Änderungen übernehmen können, finden Sie unter [the section called “Übergeben eines Datenmodells”](#).

## So übertragen Sie Datenmodelle an Amazon Keyspaces und Apache Cassandra

In diesem Abschnitt erfahren Sie, wie Sie fertige Datenmodelle in Amazon Keyspaces- und Apache Cassandra-Cluster übertragen. Dieser Prozess erstellt automatisch die serverseitigen Ressourcen für Schlüsselräume und Tabellen auf der Grundlage der Einstellungen, die Sie im Datenmodell definiert haben.

The screenshot shows the NoSQL Workbench Visualizer interface. On the left, there is a sidebar with navigation options like 'AWS database catalog', 'Amazon Keyspaces', 'Data modeler', 'Visualizer', 'Documentation', and 'Email us'. The main area is titled 'Visualizer' and shows a 'Data model' for 'Airline Operations Data' in the 'flights' Keyspace. A table named 'routes' is selected, and its schema is displayed in a grid view. The table has a primary key consisting of 'airline\_id (text)', 'origin\_id (text)', 'destination\_id (text)', and 'stops (text)'. The 'Non-key columns' are 'origin\_airport (text)', 'destination\_airport (text)', and 'equipment'. The table contains three rows of data. At the bottom of the interface, there are three buttons: 'Aggregate view', 'Commit to Amazon Keyspaces' (highlighted with a red box), and 'Commit to Apache Cassandra'.

| Primary key       |                  |                       |              | Non-key columns                  |                              |           |
|-------------------|------------------|-----------------------|--------------|----------------------------------|------------------------------|-----------|
| Partition key     |                  | Clustering columns    |              |                                  |                              |           |
| airline_id (text) | origin_id (text) | destination_id (text) | stops (text) | origin_airport (text)            | destination_airport (text)   | equipment |
| acme_airlines     | EWR              | MCI                   | 1            | Newark Liberty International     | Kansas City International    | 737       |
| trusted_airlines  | PHX              | MIC                   | 2            | Phoenix Sky Harbor International | Kansas City International    | 737       |
| freedom_airlines  | SFO              | EWR                   | 1            | San Francisco International      | Newark Liberty International | 747       |

## Themen

- [Bevor Sie beginnen](#)
- [Verbindung zu Amazon Keyspaces mit dienstspezifischen Anmeldeinformationen herstellen](#)
- [Verbindung zu Amazon Keyspaces mit AWS Identity and Access Management \(IAM-\) Anmeldeinformationen herstellen](#)
- [Verwenden einer gespeicherten Verbindung](#)
- [Ich bekenne mich zu Apache Cassandra](#)

## Bevor Sie beginnen

Amazon Keyspaces erfordert die Verwendung von Transport Layer Security (TLS), um Verbindungen mit Kunden zu sichern. Um mithilfe von TLS eine Verbindung zu Amazon Keyspaces herzustellen, müssen Sie die folgende Aufgabe ausführen, bevor Sie beginnen können.

- Laden Sie das digitale Starfield-Zertifikat mit dem folgenden Befehl herunter und speichern Sie es `sf-class2-root.crt` lokal oder in Ihrem Home-Verzeichnis.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -O
```

#### Note

Sie können das digitale Zertifikat von Amazon auch verwenden, um eine Verbindung zu Amazon Keyspaces herzustellen. Sie können dies auch weiterhin tun, wenn Ihr Kunde erfolgreich eine Verbindung zu Amazon Keyspaces herstellt. Das Starfield-Zertifikat bietet zusätzliche Abwärtskompatibilität für Kunden, die ältere Zertifizierungsstellen verwenden.

```
curl https://certs.secureserver.net/repository/sf-class2-root.crt -O
```

Nachdem Sie die Zertifikatsdatei gespeichert haben, können Sie eine Verbindung zu Amazon Keyspaces herstellen. Eine Möglichkeit besteht darin, mithilfe dienstspezifischer Anmeldeinformationen eine Verbindung herzustellen. Dienstspezifische Anmeldeinformationen sind ein Benutzername und ein Passwort, die einem bestimmten IAM-Benutzer zugeordnet sind und nur mit dem angegebenen Dienst verwendet werden können. Die zweite Option besteht darin, eine Verbindung mit IAM-Anmeldeinformationen herzustellen, die den [AWSSignature Version 4-Prozess \(SIGv4\)](#) verwenden. Weitere Informationen zu diesen beiden Optionen finden Sie unter [the section called “Anmeldeinformationen erstellen”](#).

Informationen zum Herstellen einer Verbindung mit dienstspezifischen Anmeldeinformationen finden Sie unter [the section called “Verbindung mit dienstspezifischen Anmeldeinformationen herstellen”](#).

Informationen zum Herstellen einer Verbindung mit IAM-Anmeldedaten finden Sie unter [the section called “Verbindung mit IAM-Anmeldeinformationen herstellen”](#).

## Verbindung zu Amazon Keyspaces mit dienstspezifischen Anmeldeinformationen herstellen

In diesem Abschnitt wird gezeigt, wie Sie dienstspezifische Anmeldeinformationen verwenden, um das Datenmodell, das Sie mit NoSQL Workbench erstellt oder bearbeitet haben, zu übertragen.

1. Um eine neue Verbindung mit dienstspezifischen Anmeldeinformationen herzustellen, wählen Sie die Registerkarte Mit Benutzername und Passwort verbinden.
  - Bevor Sie beginnen, müssen Sie dienstspezifische Anmeldeinformationen mithilfe des unter beschriebenen Verfahrens erstellen. [the section called “Dienstspezifische Anmeldeinformationen”](#)

Nachdem Sie die dienstspezifischen Anmeldeinformationen erhalten haben, können Sie mit der Einrichtung der Verbindung fortfahren. Fahren Sie mit einem der folgenden Schritte fort:

- Benutzername — Geben Sie den Benutzernamen ein.
- Passwort — Geben Sie das Passwort ein.
- AWS-Region— Verfügbare Regionen finden Sie unter [the section called “Service-Endpunkte”](#).
- Port — Amazon Keyspaces verwendet Port 9142.

Alternativ können Sie gespeicherte Anmeldeinformationen aus einer Datei importieren.

2. Wählen Sie Commit, um Amazon Keyspaces mit dem Datenmodell zu aktualisieren.



## Commit to Amazon Keyspaces

**i** On this page, you can create server-side resources such as keyspaces and tables for the chosen data model.

< Use saved connections    Connect by using IAM credentials    Connect by using user name >

**i** You can generate service-specific credentials to allow your users to access Amazon Keyspaces using AWS Management Console or AWS CLI.

[How to generate Amazon Keyspaces credentials](#)

\* User Name

anika

\* Password

.....



\* AWS Region

us-east-1



\* Port

9142

OR

[Import from credential file](#)

Cancel

Reset

Commit

## Verbindung zu Amazon Keyspaces mit AWS Identity and Access Management (IAM-) Anmeldeinformationen herstellen

In diesem Abschnitt wird gezeigt, wie Sie IAM-Anmeldedaten verwenden, um das mit NoSQL Workbench erstellte oder bearbeitete Datenmodell zu übertragen.

1. Um eine neue Verbindung mit IAM-Anmeldeinformationen zu erstellen, wählen Sie die Registerkarte Mit IAM-Anmeldeinformationen verbinden.
  - Bevor Sie beginnen, müssen Sie IAM-Anmeldeinformationen mit einer der folgenden Methoden erstellen.
    - Verwenden Sie für den Zugriff auf die Konsole Ihren IAM-Benutzernamen und Ihr Passwort, um sich [AWS Management Console](#) von der IAM-Anmeldeseite aus bei der anzumelden. Informationen zu AWS Sicherheitsanmeldeinformationen, einschließlich des programmatischen Zugriffs und Alternativen zu langfristigen Anmeldeinformationen, finden Sie unter [AWSSicherheitsanmeldeinformationen](#) im IAM-Benutzerhandbuch. Einzelheiten zur Anmeldung bei Ihrem AWS-Konto finden Sie [im AWS-AnmeldungBenutzerhandbuch unter So melden Sie sich an AWS](#).
    - Für CLI-Zugriff benötigen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel. Verwenden Sie möglichst temporäre Anmeldeinformationen anstelle langfristiger Zugriffsschlüssel. Temporäre Anmeldeinformationen bestehen aus einer Zugriffsschlüssel-ID, einem geheimen Zugriffsschlüssel und einem Sicherheits-Token, das angibt, wann die Anmeldeinformationen ablaufen. Weitere Informationen finden Sie unter [Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen](#) im IAM-Benutzerhandbuch.
    - Für den API-Zugriff benötigen Sie eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel. Verwenden Sie IAM-Benutzer-Zugriffsschlüssel anstelle von Root-Benutzer des AWS-Kontos-Zugriffsschlüsseln. Weitere Information über IAM-Zugriffsschlüssel finden Sie unter [Verwalten von Zugriffsschlüsseln für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

Weitere Informationen finden Sie unter [Zugriffsschlüssel für IAM-Benutzer verwalten](#).

Nachdem Sie die IAM-Anmeldeinformationen erhalten haben, können Sie mit der Einrichtung der Verbindung fortfahren.

- Verbindungsname — Der Name der Verbindung.
  - AWS-Region— Verfügbare Regionen finden Sie unter [the section called “Service-Endpunkte”](#).
  - Zugangsschlüssel-ID — Geben Sie die Zugangsschlüssel-ID ein.
  - Geheimer Zugriffsschlüssel — Geben Sie den geheimen Zugangsschlüssel ein.
  - Port — Amazon Keyspaces verwendet Port 9142.
  - AWSöffentliches Zertifikat — Zeigen Sie auf das AWS Zertifikat, das im ersten Schritt heruntergeladen wurde.
  - Verbindung aufrechterhalten — Aktivieren Sie dieses Kontrollkästchen, wenn Sie die AWS Verbindungsgeheimnisse lokal speichern möchten.
2. Wählen Sie Commit, um Amazon Keyspaces mit dem Datenmodell zu aktualisieren.

**i** On this page, you can create server-side resources such as keyspaces and tables for the chosen data model.

< Use saved connections Connect by using IAM credentials Connect by using user name >

\* Connection name

Connection 2

\* AWS Region

us-east-2

\* Access key ID

AKIAIOSFODNN7EXAMPLE

\* Secret access key

.....

\* Port

9142

\* AWS public certificate

⬇ Choose AWS public certificate AmazonRootCA1.pem

Choose an AWS public certificate for a Signature Version 4–based connection to Amazon Keyspaces. **i**

Persist connection

If you select this check box, AWS connection secrets will be persisted in

C:\Users\la...\.aws\credentials

Cancel

Reset

Commit

## Verwenden einer gespeicherten Verbindung

Wenn Sie zuvor eine Verbindung zu Amazon Keyspaces eingerichtet haben, können Sie diese als Standardverbindung verwenden, um Datenmodelländerungen zu übertragen. Wählen Sie den Tab Gespeicherte Verbindungen verwenden und fahren Sie mit dem Übertragen der Updates fort.

### Commit to Amazon Keyspaces



On this page, you can create server-side resources such as keyspaces and tables for the chosen data model.

< **Use saved connections**    Connect by using IAM credentials    Connect by using user name : >

\* Saved connections

default



\* Port

9142

\* AWS public certificate

[Choose AWS public certificate](#)

AmazonRootCA1.pem

Choose an AWS public certificate for a Signature Version 4–based connection to Amazon Keyspaces.


Cancel

Reset

Commit

## Ich bekenne mich zu Apache Cassandra

In diesem Abschnitt erfahren Sie, wie Sie die Verbindungen zu einem Apache Cassandra-Cluster herstellen, um das mit NoSQL Workbench erstellte oder bearbeitete Datenmodell zu übertragen.

 Note

Nur Datenmodelle, die mit Apache Cassandra-Clustern erstellt wurden `SimpleStrategy` oder für Apache Cassandra-Cluster übernommen werden `NetworkTopologyStrategy` können. Um die Replikationsstrategie zu ändern, bearbeiten Sie den Schlüsselraum im Datenmodellierer.

- **Benutzername** — Geben Sie den Benutzernamen ein, wenn die Authentifizierung im Cluster aktiviert ist.
  - **Passwort** — Geben Sie das Passwort ein, wenn die Authentifizierung im Cluster aktiviert ist.
  - **Kontaktstellen** — Geben Sie die Kontaktpunkte ein.
  - **Lokales Rechenzentrum** — Geben Sie den Namen des lokalen Rechenzentrums ein.
  - **Port** — Die Verbindung verwendet Port 9042.
- Wählen Sie **Commit**, um den Apache Cassandra-Cluster mit dem Datenmodell zu aktualisieren.

## Commit to Apache Cassandra



Configure the connection to the Apache Cassandra cluster so that you can commit your data model to the database, including keyspace, tables, and sample data. The user name and password are not required, and are necessary only if authentication is enabled on the cluster

User name

Password

\* Contact points

[+ Add another contact point](#)

\* Local data center

\* Port

Cancel

Reset

Commit

# Beispieldatenmodelle in NoSQL Workbench

Auf der Homepage für den Modeler und Visualizer werden eine Reihe von Beispielmustern angezeigt, die im Lieferumfang von NoSQL Workbench enthalten sind. In diesem Abschnitt werden diese Modelle und ihre möglichen Verwendungsmöglichkeiten beschrieben.

Themen

- [Mitarbeiterdatenmodell](#)
- [Datenmodell für Kreditkartentransaktionen](#)
- [Datenmodell für den Flugbetrieb](#)

## Mitarbeiterdatenmodell

Dieses Datenmodell stellt ein Amazon Keyspaces-Schema für eine Mitarbeiterdatenbankanwendung dar.

Anwendungen, die auf Mitarbeiterinformationen für ein bestimmtes Unternehmen zugreifen, können dieses Datenmodell verwenden.

Die von diesem Datenmodell unterstützten Zugriffsmuster sind:

- Abruf eines Mitarbeiterdatensatzes mit einer bestimmten ID.
- Abruf eines Mitarbeiterdatensatzes mit einer bestimmten ID und Abteilung.
- Abruf eines Mitarbeiterdatensatzes mit einer bestimmten ID und einem bestimmten Namen.

## Datenmodell für Kreditkartentransaktionen

Dieses Datenmodell stellt ein Amazon Keyspaces-Schema für Kreditkartentransaktionen in Einzelhandelsgeschäften dar.

Die Speicherung von Kreditkartentransaktionen hilft Geschäften nicht nur bei der Buchhaltung, sondern hilft den Filialleitern auch dabei, Kaufrends zu analysieren, was ihnen bei Prognosen und Planungen helfen kann.

Die von diesem Datenmodell unterstützten Zugriffsmuster sind:

- Abruf von Transaktionen nach Kreditkartennummer, Monat und Jahr sowie Datum.



- Abruf von Transaktionen nach Kreditkartennummer, Kategorie und Datum.
- Abruf von Transaktionen nach Kategorie, Standort und Kreditkartennummer.
- Abruf von Transaktionen nach Kreditkartennummer und Streitstatus.

## Datenmodell für den Flugbetrieb

Dieses Datenmodell zeigt Daten über Flugzeugflüge, einschließlich Flughäfen, Fluggesellschaften und Flugrouten.

Die wichtigsten Komponenten der Amazon Keyspaces-Modellierung, die demonstriert werden, sind Schlüsselwertpaare, Datenspeicher mit großen Spalten, zusammengesetzte Schlüssel und komplexe Datentypen wie Karten zur Veranschaulichung gängiger NoSQL-Datenzugriffsmuster.

Die von diesem Datenmodell unterstützten Zugriffsmuster sind:

- Abruf von Strecken, die von einer bestimmten Fluggesellschaft an einem bestimmten Flughafen ausgehen.
- Abruf von Routen mit einem bestimmten Zielflughafen.
- Abruf von Flughäfen mit direkten Flügen.
- Abruf von Flughafen- und Airline-Details.

## Versionsverlauf für NoSQL-Workbench

In der folgenden Tabelle werden die wichtigen Änderungen in den einzelnen Versionen der Client-Workbench NoSQL-Workbench NoSQL-Workbench NoSQL-Workbench NoSQL-Workbench beschrieben.

| Änderung                                     | Beschreibung                                                             | Datum            |
|----------------------------------------------|--------------------------------------------------------------------------|------------------|
| NoSQL-Workbench für Amazon Keyspaces — GA.   | NoSQL-Workbench für Amazon Keyspaces ist allgemein verfügbar.            | 28. Oktober 2020 |
| NoSQL-Workbench (Vorversion) veröffentlicht. | NoSQL Workbench ist eine clientseitige Anwendung , mit der Sie nichtrela | 5. Oktober 2020  |

| Änderung | Beschreibung                                                                                                                                                                                                                                             | Datum |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
|          | <p>tionale Datenmodelle für Amazon Keyspaces einfacher entwerfen und visualisieren können. NoSQL-Workbench sind für Windows, macOS und Linux verfügbar. Weitere Informationen finden Sie unter <a href="#">NoSQL-Workbench für Amazon Keyspaces</a>.</p> |       |

# Multiregionale Replikation für Amazon Keyspaces (für Apache Cassandra)

Sie können die Multi-Region-Replikation von Amazon Keyspaces verwenden, um Ihre Daten mit automatisierter, vollständig AWS-Regionen verwalteter, aktiv-aktiver Replikation über die Ihrer Wahl zu replizieren. Bei der Aktiv-Aktiv-Replikation ist jede Region in der Lage, Lese- und Schreibvorgänge isoliert durchzuführen. Sie können sowohl die Verfügbarkeit als auch die Ausfallsicherheit durch regionale Verschlechterungen verbessern und gleichzeitig von lokalen Lese- und Schreibvorgängen mit niedriger Latenz für globale Anwendungen profitieren.

Bei der Multi-Region-Replikation repliziert Amazon Keyspaces asynchron Daten zwischen Regionen, und Daten werden in der Regel innerhalb einer Sekunde über Regionen verteilt. Außerdem haben Sie bei der Multi-Region-Replikation nicht mehr die schwierige Arbeit, Konflikte zu lösen und Datendivergenzprobleme zu korrigieren, sodass Sie sich auf Ihre Anwendung konzentrieren können.

Standardmäßig repliziert Amazon Keyspaces Daten in drei [Availability Zones](#) innerhalb derselben AWS-Region für Haltbarkeit und hohe Verfügbarkeit. Mit der Multi-Region-Replikation können Sie multiregionale Keyspaces erstellen, die Ihre Tabellen in bis zu sechs verschiedenen geografischen Regionen AWS-Regionen Ihrer Wahl replizieren.

## Themen

- [Vorteile der Verwendung der Multi-Region-Replikation](#)
- [Kapazitätsmodi und Preise](#)
- [So funktioniert die regionsübergreifende Replikation in Amazon Keyspaces](#)
- [Nutzungshinweise für Amazon Keyspaces Multi-Region Replication](#)
- [So verwenden Sie die Multi-Region-Replikation](#)

## Vorteile der Verwendung der Multi-Region-Replikation

Die Replikation in mehreren Regionen bietet die folgenden Vorteile.

- Globale Lese- und Schreibvorgänge mit Latenz im einstelligen Millisekundenbereich – In Amazon Keyspaces ist die Replikation aktiv-aktiv. Sie können Lese- und Schreibvorgänge lokal aus den -Regionen bereitstellen, die Ihren Kunden am nächsten liegen, mit Latenz im einstelligen

Millisekundenbereich in jeder Größenordnung. Sie können Amazon-Keyspaces-Tabellen für globale Anwendungen verwenden, die überall auf der Welt eine schnelle Reaktionszeit benötigen.

- Verbesserte Geschäftskontinuität und Schutz vor Beeinträchtigungen in einer einzelnen Region – Mit der Multi-Region-Replikation können Sie sich von der Verschlechterung in einer einzelnen Region erholen, AWS-Region indem Sie Ihre Anwendung in eine andere Region in Ihrem multiregionalen Keyspace umleiten. Da Amazon Keyspaces eine Aktiv-Aktiv-Replikation bietet, hat dies keine Auswirkungen auf Ihre Lese- und Schreibvorgänge.

Amazon Keyspaces verfolgt alle Schreibvorgänge, die in Ihrem multiregionalen Keyspace durchgeführt, aber nicht in alle Replikatregionen weitergegeben wurden. Nachdem die Region wieder online ist, synchronisiert Amazon Keyspaces automatisch alle fehlenden Änderungen, sodass Sie die Wiederherstellung ohne Anwendungsauswirkungen durchführen können.

- Hochgeschwindigkeitsreplikation über Regionen hinweg – Die Multi-Region-Replikation verwendet eine schnelle, speicherbasierte physische Replikation von Daten über Regionen hinweg, mit einer Replikationsverzögerung von in der Regel weniger als 1 Sekunde.

Die Replikation in Amazon Keyspaces hat nur geringe bis gar keine Auswirkungen auf Ihre Datenbankabfragen, da sie keine Rechenressourcen für Ihre Anwendung freigibt. Das bedeutet, dass Sie Anwendungsfälle mit hohem Schreibdurchsatz oder Anwendungsfälle mit plötzlichen Spitzen oder Spitzen im Durchsatz bewältigen können, ohne dass sich dies auf die Anwendung auswirkt.

- Konsistenz und Konfliktlösung – Alle Änderungen an Daten in einer beliebigen Region werden in die anderen Regionen in einem multiregionalen Keyspace repliziert. Wenn Anwendungen dieselben Daten in verschiedenen Regionen gleichzeitig aktualisieren, können Konflikte auftreten.

Um letztendliche Konsistenz zu gewährleisten, verwendet Amazon Keyspaces Zeitstempel auf Zellebene und ein letzter Writer gewinnt die Abstimmung zwischen gleichzeitigen Updates. Die Konfliktlösung wird vollständig verwaltet und erfolgt im Hintergrund ohne Auswirkungen auf die Anwendung.

Weitere Informationen zu unterstützten Konfigurationen und Funktionen finden Sie unter [the section called “Nutzungshinweise”](#).

## Kapazitätsmodi und Preise

Für einen multiregionalen Keyspace können Sie entweder den On-Demand-Kapazitätsmodus oder den Modus bereitgestellter Kapazität verwenden. Weitere Informationen finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).

Im On-Demand-Modus werden Ihnen 1,25 Schreibenanforderungseinheiten (WRUs) in Rechnung gestellt, um bis zu 1 KB Daten pro Zeile zu schreiben. Schreibvorgänge werden Ihnen in jeder Region Ihres multiregionalen Keyspace in Rechnung gestellt. Beispielsweise erfordert das Schreiben einer Zeile mit 3 KB Daten in einem multiregionalen Keyspace mit zwei Regionen 7,5 WRUs:  $3 * 1,25 * 2 = 7,5$  WRUs. Darüber hinaus erfordern Schreibvorgänge, die sowohl statische als auch nichtstatische Daten enthalten, zusätzliche Schreibvorgänge.

Für den Bereitstellungsmodus werden Ihnen 1,25 Schreibkapazitätseinheiten (WCUs) in Rechnung gestellt, um bis zu 1 KB Daten pro Zeile zu schreiben. Schreibvorgänge werden Ihnen in jeder Region Ihres multiregionalen Keyspace in Rechnung gestellt. Beispielsweise erfordert das Schreiben einer Zeile mit 3 KB Daten pro Sekunde in einem multiregionalen Keyspace mit zwei Regionen 7,5 WCUs:  $3 * 1,25 * 2 = 7,5$  WCUs. Darüber hinaus erfordern Schreibvorgänge, die sowohl statische als auch nichtstatische Daten enthalten, zusätzliche Schreibvorgänge.

Weitere Informationen zu Preisen finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\) – Preise](#).

## So funktioniert die regionsübergreifende Replikation in Amazon Keyspaces

Dieser Abschnitt bietet einen Überblick darüber, wie Amazon Keyspaces Multi-Region Replication funktioniert. Weitere Informationen zu den Preisen finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\) — Preise](#).

### Themen

- [So funktioniert die regionsübergreifende Replikation in Amazon Keyspaces](#)
- [Lösung von Konflikten bei der Replikation mehrerer Regionen](#)
- [Notfallwiederherstellung bei Replikation in mehreren Regionen](#)
- [IAM-Berechtigungen sind erforderlich, um Schlüsselräume und Tabellen für mehrere Regionen zu erstellen](#)

- [Replikation und Integration mit Wiederherstellung in mehreren Regionen \(PITR\) point-in-time](#)
- [Replikation und Integration mit Diensten in mehreren Regionen AWS](#)

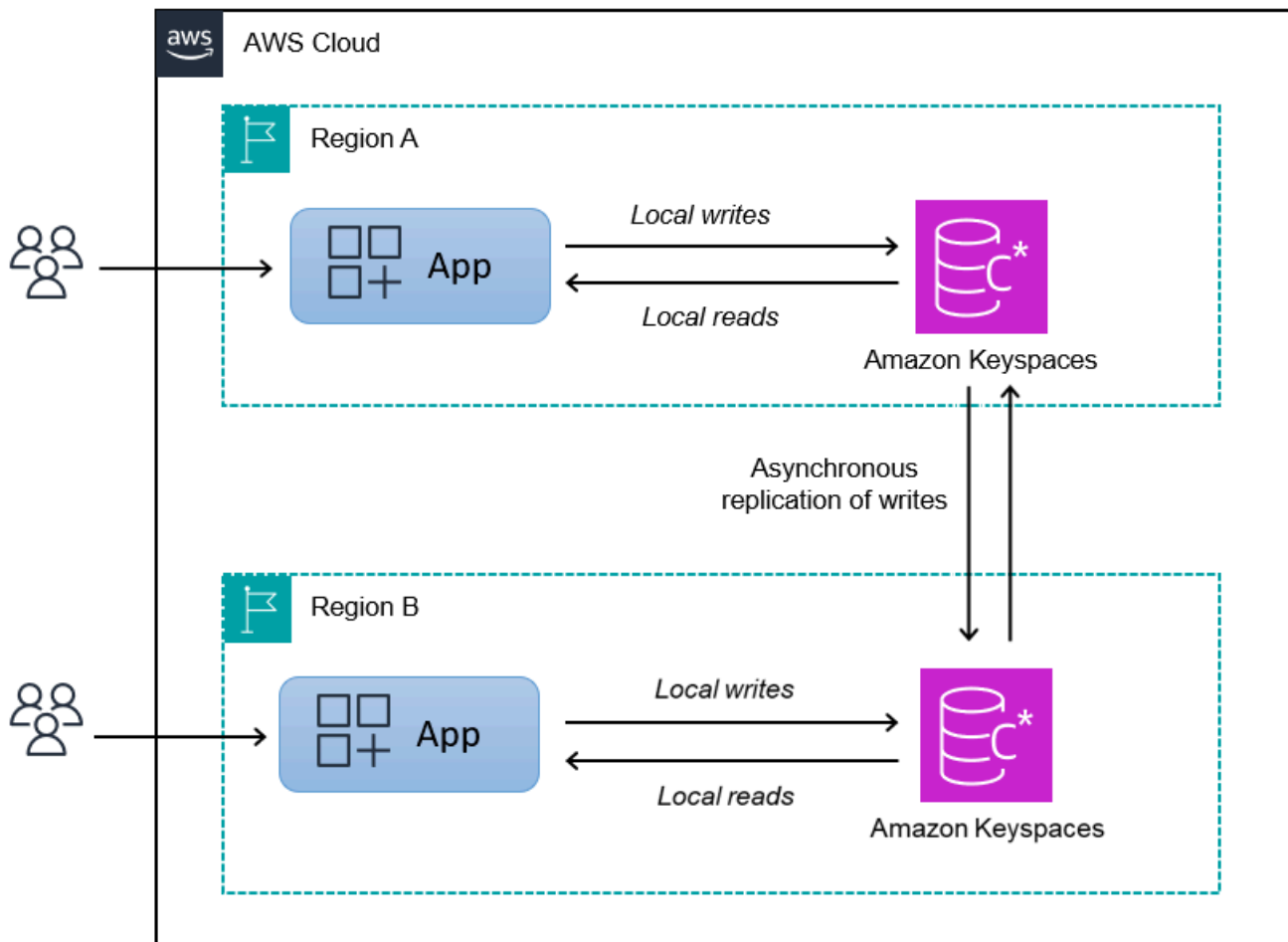
## So funktioniert die regionsübergreifende Replikation in Amazon Keyspaces

Amazon Keyspaces Multi-Region Replication implementiert eine Datenausfallsicherheitsarchitektur, die Ihre Daten über unabhängige und geografisch verteilte Standorte verteilt. AWS-Regionen Es verwendet eine aktiv-aktive Replikation, die lokal eine geringe Latenz bietet, sodass jede Region Lese- und Schreibvorgänge isoliert durchführen kann.

Wenn Sie einen Amazon Keyspaces-Schlüsselraum für mehrere Regionen erstellen, können Sie bis zu fünf zusätzliche Regionen auswählen, in die die Daten repliziert werden sollen. Jede Tabelle, die Sie in einem Schlüsselraum mit mehreren Regionen erstellen, besteht aus mehreren Replikattabellen (eine pro Region), die Amazon Keyspaces als eine Einheit betrachtet.

Jedes Replikat hat den gleichen Tabellennamen und das gleiche Primärschlüsselschema. Wenn eine Anwendung Daten in eine lokale Tabelle in einer Region schreibt, werden die Daten dauerhaft unter Verwendung der Konsistenzstufe geschrieben. LOCAL\_QUORUM Amazon Keyspaces repliziert die Daten automatisch asynchron in die anderen Replikationsregionen. Die Verzögerung bei der Replikation zwischen den Regionen beträgt in der Regel weniger als eine Sekunde und hat keinen Einfluss auf die Leistung oder den Durchsatz Ihrer Anwendung.

Nachdem die Daten geschrieben wurden, können Sie sie aus der Tabelle mit mehreren Regionen in einer anderen Replikationsregion mit den LOCAL\_ONE/LOCAL\_QUORUM Konsistenzstufen lesen. Weitere Informationen zu unterstützten Konfigurationen und Funktionen finden Sie unter [the section called "Nutzungshinweise"](#).



## Lösung von Konflikten bei der Replikation mehrerer Regionen

Amazon Keyspaces Multi-Region Replication wird vollständig verwaltet, was bedeutet, dass Sie keine Replikationsaufgaben wie das regelmäßige Ausführen von Reparaturvorgängen zur Behebung von Datensynchronisierungsproblemen durchführen müssen. Amazon Keyspaces überwacht die Datenkonsistenz zwischen Tabellen in verschiedenen Tabellen, AWS-Regionen indem es Konflikte erkennt und repariert und Replikate automatisch synchronisiert.

Amazon Keyspaces verwendet die Methode des Datenabgleichs nach dem letzten Writer-Wins-Verfahren. Mit diesem Konfliktlösungsmechanismus einigen sich alle Regionen in einem Schlüsselraum mit mehreren Regionen auf das neueste Update und nähern sich einem Zustand an, in dem sie alle über identische Daten verfügen. Der Abgleichprozess hat keine Auswirkungen auf die Anwendungsleistung. Zur Unterstützung der Konfliktlösung werden clientseitige Zeitstempel für Tabellen mit mehreren Regionen automatisch aktiviert und können nicht deaktiviert werden. Weitere Informationen finden Sie unter [ClientZeitstempel](#).

## Notfallwiederherstellung bei Replikation in mehreren Regionen

Mit Amazon Keyspaces Multi-Region Replication werden Schreibvorgänge asynchron in jeder Region repliziert. Im seltenen Fall, dass eine einzelne Region beeinträchtigt wird oder ausfällt, hilft Ihnen Multi-Region-Replikation dabei, Ihre Anwendung nach einem Notfall wiederherzustellen, ohne dass dies Auswirkungen auf Ihre Anwendung hat. Die Wiederherstellung nach einem Notfall wird in der Regel anhand von Werten für Recovery Time Objective (RTO) und Recovery Point Objective (RPO) gemessen.

**Ziel der Wiederherstellungszeit** — Die Zeit, die ein System benötigt, um nach einem Notfall wieder betriebsbereit zu sein. RTO misst die Menge an Ausfallzeiten, die Ihr Workload tolerieren kann, gemessen an der Zeit. Bei Disaster-Recovery-Plänen, die Replikation über mehrere Regionen verwenden, um ein Failover auf eine Region durchzuführen, die nicht betroffen ist, kann die RTO fast Null sein. Die RTO hängt davon ab, wie schnell Ihre Anwendung den Fehler erkennen und den Datenverkehr in eine andere Region umleiten kann.

**Ziel des Wiederherstellungspunkts** — Die Datenmenge, die verloren gehen kann (gemessen in der Zeit). Bei Disaster-Recovery-Plänen, bei denen mithilfe der Replikation mehrerer Regionen ein Failover auf eine Region erfolgt, die nicht betroffen ist, beträgt der RPO-Wert in der Regel einstellige Sekundenzahlen. Das RPO ist durch die Replikationslatenz auf das Failover-Zielreplikat begrenzt.

Im Falle eines regionalen Fehlers oder einer Verschlechterung müssen Sie keine sekundäre Region heraufstufen oder Datenbank-Failover-Verfahren durchführen, da die Replikation in Amazon Keyspaces aktiv und aktiv ist. Stattdessen können Sie Amazon Route 53 verwenden, um Ihre Anwendung an die nächstgelegene gesunde Region weiterzuleiten. Weitere Informationen zu Route 53 finden Sie unter [Was ist Amazon Route 53?](#) .

Wenn eine einzelne Komponente isoliert oder beeinträchtigt AWS-Region wird, kann Ihre Anwendung den Datenverkehr mithilfe von Route 53 in eine andere Region umleiten, um Lese- und Schreibvorgänge in einer anderen Replikattabelle durchzuführen. Sie können auch eine benutzerdefinierte Geschäftslogik anwenden, um zu bestimmen, wann Anfragen in andere Regionen umgeleitet werden sollen. Ein Beispiel dafür ist, dass Ihre Anwendung auf die verschiedenen verfügbaren Endpunkte aufmerksam macht.

Wenn die Region wieder online ist, setzt Amazon Keyspaces die Weitergabe aller ausstehenden Schreibvorgänge aus dieser Region an die Replikattabellen in anderen Regionen fort. Außerdem wird das Übertragen von Schreibvorgängen aus anderen Replikattabellen in die Region fortgesetzt, die jetzt wieder online ist.



## IAM-Berechtigungen sind erforderlich, um Schlüsselräume und Tabellen für mehrere Regionen zu erstellen

Um Schlüsselräume und Tabellen mit mehreren Regionen erfolgreich zu erstellen, muss der IAM-Prinzipal in der Lage sein, eine dienstbezogene Rolle zu erstellen. Bei dieser serviceverknüpften Rolle handelt es sich um eine einzigartige Art von IAM-Rolle, die von Amazon Keyspaces vordefiniert ist. Es beinhaltet alle Berechtigungen, die Amazon Keyspaces benötigt, um Aktionen in Ihrem Namen durchzuführen. Weitere Informationen zur serviceverknüpften Rolle finden Sie unter [the section called "Multiregionale Replikation"](#).

Um die serviceverknüpfte Rolle zu erstellen, die für die Multi-Region-Replikation erforderlich ist, erfordert die Richtlinie für den IAM-Prinzipal die folgenden Elemente:

- `iam:CreateServiceLinkedRole`— Die Aktion, die der Principal ausführen kann.
- `arn:aws:iam::*:role/aws-service-role/replication.cassandra.amazonaws.com/AWSServiceRoleForKeyspacesReplication`— Die Ressource, auf der die Aktion ausgeführt werden kann.
- `iam:AWSServiceName": "replication.cassandra.amazonaws.com`— Der einzige AWS Service, an den diese Rolle angehängt werden kann, ist Amazon Keyspaces.

Im Folgenden finden Sie ein Beispiel für die Richtlinie, die einem Prinzipal die erforderlichen Mindestberechtigungen zum Erstellen von Schlüsselräumen und Tabellen für mehrere Regionen gewährt.

```
{
 "Effect": "Allow",
 "Action": "iam:CreateServiceLinkedRole",
 "Resource": "arn:aws:iam::*:role/aws-service-role/
replication.cassandra.amazonaws.com/AWSServiceRoleForKeyspacesReplication",
 "Condition": {"StringLike": {"iam:AWSServiceName":
"replication.cassandra.amazonaws.com"}}
}
```

Zusätzliche IAM-Berechtigungen für Schlüsselräume und Tabellen mit mehreren Regionen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Keyspaces \(für Apache Cassandra\)](#) in der Service Authorization Reference.

## Replikation und Integration mit Wiederherstellung in mehreren Regionen (PITR) point-in-time

Die point-in-time P-Wiederherstellung wird in Tabellen mit mehreren Regionen unterstützt. Um eine Tabelle mit mehreren Regionen erfolgreich mit PITR wiederherzustellen, müssen die folgenden Bedingungen erfüllt sein.

- Die Quell- und die Zieltabelle müssen als Tabellen mit mehreren Regionen konfiguriert sein.
- Die Replikationsregionen für den Schlüsselraum der Quelltable und für den Schlüsselraum der Zieltabelle müssen identisch sein.

Sie können die Restore-Anweisung von jeder Region aus ausführen, in der die Quelltable verfügbar ist. Amazon Keyspaces stellt die Zieltabelle in jeder Region automatisch wieder her. Weitere Informationen zu PITR finden Sie unter [the section called “Funktionsweise”](#).

## Replikation und Integration mit Diensten in mehreren Regionen AWS

Sie können die Replikationsleistung zwischen Tabellen in verschiedenen Tabellen mithilfe AWS-Regionen von CloudWatch Amazon-Metriken überwachen. Die folgende Metrik ermöglicht die kontinuierliche Überwachung von Schlüsselräumen in mehreren Regionen.

- `ReplicationLatency`— Diese Metrik misst die Zeitupdates, inserts die für die Replikation oder deletes von einer Replikattabelle in eine andere Replikattabelle in einem Schlüsselraum mit mehreren Regionen benötigt wurde.

Weitere Informationen zur Überwachung von Metriken finden Sie unter. CloudWatch [the section called “Überwachung mit CloudWatch”](#)

## Nutzungshinweise für Amazon Keyspaces Multi-Region Replication

Beachten Sie Folgendes, wenn Sie Multi-Region-Replikation mit Amazon Keyspaces verwenden.

- Sie können bis zu sechs der [verfügbaren](#) öffentlichen Objekte auswählen. AWS-Regionen AWS GovCloud (US) Regions, Regionen in China, AWS-Regionen [die standardmäßig deaktiviert sind](#), werden nicht unterstützt.
- Wählen Sie die Replikationsregionen für den Schlüsselraum sorgfältig aus, da Sie sie später weder hinzufügen noch entfernen können.

- Schließen Sie das Tabellenschema ab, bevor Sie eine Tabelle mit mehreren Regionen erstellen, da Sie später keine neuen Spalten hinzufügen können.
- Verwenden Sie für die Verschlüsselung im Ruhezustand einen AWS eigenen Schlüssel. Vom Kunden verwaltete Schlüssel werden für Tabellen mit mehreren Regionen nicht unterstützt. Weitere Informationen finden Sie unter

[the section called “Funktionsweise”](#).

- Wenn Sie das bereitgestellte Kapazitätsmanagement mit Amazon Keyspaces Auto Scaling verwenden, stellen Sie sicher, dass Sie die Amazon Keyspaces-API-Operationen verwenden, um Ihre Tabellen mit mehreren Regionen zu erstellen und zu konfigurieren. Die zugrunde liegenden API-Operationen für Application Auto Scaling, die Amazon Keyspaces in Ihrem Namen aufruft, verfügen nicht über Funktionen für mehrere Regionen.

Weitere Informationen finden Sie unter [the section called “So verwenden Sie die Multi-Region-Replikation”](#). Weitere Informationen zur Schätzung des Schreibkapazitätsdurchsatzes von bereitgestellten Tabellen mit mehreren Regionen finden Sie unter [the section called “Multiregionale Tabellen”](#)

- Entscheiden Sie, ob die Tabelle Time to Live (TTL) benötigt. Sie können es später nicht mehr einschalten. Weitere Informationen finden Sie unter [Ablaufende ende Daten mit Time to Live](#).
- Obwohl Daten automatisch über die ausgewählten Regionen einer Tabelle mit mehreren Regionen repliziert werden, gibt die Abfrage nur lokale Informationen zurück, wenn ein Client eine Verbindung zu einem Endpunkt in einer Region herstellt und die `system.peers` Tabelle abfragt. Das Abfrageergebnis sieht für den Client wie ein einzelner Rechenzentrumscluster aus.
- Amazon Keyspaces Multi-Region Replication ist asynchron und unterstützt LOCAL\_QUORUM Konsistenz bei Schreibvorgängen. LOCAL\_QUORUM Konsistenz erfordert, dass eine Aktualisierung einer Zeile dauerhaft auf zwei Replikaten in der lokalen Region gespeichert wird, bevor der Client erfolgreich zurückgemeldet wird. Die Weitergabe von Schreibvorgängen in die replizierte Region (oder Regionen) erfolgt dann asynchron.

Amazon Keyspaces Multi-Region Replication unterstützt keine synchrone Replikation oder Konsistenz. QUORUM

- Wenn Sie einen Schlüsselraum oder eine Tabelle mit mehreren Regionen erstellen, werden alle Tags, die Sie während des Erstellungsprozesses definieren, automatisch auf alle Schlüsselräume und Tabellen in allen Regionen angewendet. Wenn Sie die vorhandenen Tags mit ALTER KEYSPACE oder ändern ALTER TABLE, wird die Aktualisierung nur auf den Schlüsselraum oder die Tabelle in der Region angewendet, in der Sie die Änderung vornehmen.

- Amazon CloudWatch stellt eine `ReplicationLatency` Metrik für jede replizierte Region bereit. Es berechnet diese Metrik, indem es eingehende Zeilen verfolgt, ihre Ankunftszeit mit ihrer ursprünglichen Schreibzeit vergleicht und einen Durchschnitt berechnet. Die Zeitangaben werden innerhalb der CloudWatch Quellregion gespeichert. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

Es kann nützlich sein, sich die durchschnittlichen und maximalen Zeiten anzusehen, um die durchschnittliche Verzögerung bei der Replikation und im schlimmsten Fall zu ermitteln. Für diese Latenz gibt es kein SLA.

- Wenn Sie eine Tabelle mit mehreren Regionen im On-Demand-Modus verwenden, stellen Sie möglicherweise eine Erhöhung der Latenz bei der asynchronen Replikation von Schreibvorgängen fest, wenn bei einem Tabellenreplikate eine neue Datenverkehrsspitze auftritt. Ähnlich wie Amazon Keyspaces die Kapazität einer On-Demand-Tabelle mit einer Region automatisch an den empfangenen Anwendungsdatenverkehr anpasst, passt Amazon Keyspaces die Kapazität einer On-Demand-Tabellenreplikation mit mehreren Regionen automatisch an den empfangenen Datenverkehr an. Der Anstieg der Replikationslatenz ist vorübergehend, da Amazon Keyspaces automatisch mehr Kapazität zuweist, wenn Ihr Datenverkehrsvolumen zunimmt. Sobald sich alle Replikate an Ihr Datenverkehrsvolumen angepasst haben, sollte sich die Replikationslatenz wieder normalisieren. Weitere Informationen finden Sie unter [the section called “Spitzenaufkommen an Datenverkehr und Skalierungseigenschaften”](#).
- Wenn Sie eine Tabelle mit mehreren Regionen im Bereitstellungsmodus verwenden und Ihre Anwendung die bereitgestellte Durchsatzkapazität überschreitet, können Fehler bei unzureichender Kapazität und eine Erhöhung der Replikationslatenz auftreten. Um sicherzustellen, dass immer genügend Lese- und Schreibkapazität für alle Tabellenreplikate in einer Tabelle mit mehreren Regionen vorhanden ist, empfehlen wir Ihnen, Amazon Keyspaces Auto Scaling zu konfigurieren. AWS-Regionen Amazon Keyspaces Auto Scaling hilft Ihnen dabei, Durchsatzkapazität für variable Workloads effizient bereitzustellen, indem die Durchsatzkapazität automatisch an den tatsächlichen Anwendungsdatenverkehr angepasst wird. Weitere Informationen finden Sie unter [the section called “So funktioniert Auto Scaling für Tabellen mit mehreren Regionen”](#).

## So verwenden Sie die Multi-Region-Replikation

Sie können multiregionale Keyspaces und Tabellen mithilfe der Amazon Keyspaces (für Apache Cassandra)-Konsole, Cassandra Query Language (CQL), des AWS SDK und der AWS Command Line Interface () erstellen und verwaltenAWS CLI.

Dieser Abschnitt enthält Beispiele für die Erstellung von multiregionalen Keyspaces und Tabellen mit der Konsole, mit CQL und mit der AWS CLI, wobei sowohl der On-Demand- als auch der Modus bereitgestellter Kapazität verwendet wird. Alle Tabellen, die in einem multiregionalen Keyspace erstellt wurden, erben automatisch die multiregionalen Einstellungen vom Keyspace.

Dieser Abschnitt enthält auch Beispiele für die Verwendung der Konsole, des CQL und des AWS CLI zum Verwalten der Auto-Scaling-Einstellungen von Amazon Keyspaces für bereitgestellte multiregionale Tabellen. Weitere Informationen zu allgemeinen Konfigurationsoptionen für die automatische Skalierung und deren Funktionsweise finden Sie unter [the section called “Verwaltung der Durchsatzkapazität mit Auto Scaling”](#).

Beachten Sie, dass Sie bei Verwendung des Modus bereitgestellter Kapazität für Tabellen mit mehreren Regionen immer Amazon Keyspaces API-Aufrufe verwenden müssen, um Auto Scaling zu konfigurieren. Dies liegt daran, dass die zugrunde liegenden API-Operationen von Application Auto Scaling nicht regionsfähig sind.

Weitere Informationen zur Schätzung des Schreibkapazitätsdurchsatzes von bereitgestellten multiregionalen Tabellen finden Sie unter [the section called “Multiregionale Tabellen”](#).

Weitere Informationen zur Amazon Keyspaces API finden Sie unter [Amazon Keyspaces API-Referenz](#).

Weitere Informationen zu unterstützten Konfigurationen und Multi-Region-Replikationsfunktionen finden Sie unter [the section called “Nutzungshinweise”](#).

Themen

- [Verwenden der Konsole zum Erstellen und Verwalten von multiregionalen Tabellen](#)
- [Verwenden von CQL zum Erstellen und Verwalten von multiregionalen Tabellen](#)
- [Verwenden der AWS CLI zum Erstellen und Verwalten von multiregionalen Tabellen](#)

## Verwenden der Konsole zum Erstellen und Verwalten von multiregionalen Tabellen

Dieser Abschnitt enthält Beispiele für die Erstellung von multiregionalen Keyspaces und Tabellen im On-Demand-Modus und im Modus bereitgestellter Kapazität mithilfe der Amazon Keyspaces (für Apache Cassandra)-Konsole. Alle Tabellen, die Sie in einem multiregionalen Keyspace erstellen, erben automatisch die multiregionalen Einstellungen vom Keyspace.

Beispiele für CQL finden Sie unter [the section called “Verwenden von CQL”](#). AWS CLI Beispiele finden Sie unter [the section called “Verwenden der AWS CLI”](#).

## Themen

- [Erstellen eines multiregionalen Keyspace \(Konsole\)](#)
- [Erstellen einer multiregionalen Tabelle mit Standardeinstellungen \(Konsole\)](#)
- [Erstellen einer multiregionalen Tabelle im Bereitstellungsmodus mit aktiviertem Auto Scaling \(Konsole\)](#)
- [Aktivieren der automatischen Skalierung für eine vorhandene multiregionale Tabelle \(Konsole\)](#)
- [Deaktivieren der automatischen Skalierung für eine Tabelle mit mehreren Regionen \(Konsole\)](#)
- [Anzeigen von Auto-Scaling-Aktivitäten von Amazon Keyspaces in der Konsole](#)

## Erstellen eines multiregionalen Keyspace (Konsole)

Gehen Sie wie folgt vor, um mithilfe der Amazon Keyspaces-Konsole einen neuen multiregionalen Keyspace zu erstellen.

So erstellen Sie einen multiregionalen Keyspace (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Keyspaces und dann Keyspace erstellen aus.
3. Geben Sie für Keyspace-Name den Namen für den Keyspace ein.
4. Im Abschnitt Multiregionale Replikation können Sie bis zu fünf zusätzliche Regionen hinzufügen, die in der Liste verfügbar sind.
5. Um den Vorgang abzuschließen, wählen Sie Keyspace erstellen aus.

### Note

Beim Erstellen eines multiregionalen Keyspace erstellt Amazon Keyspaces eine serviceverknüpfte Rolle mit dem Namen `AWSServiceRoleForAmazonKeyspacesReplication` in Ihrem Konto. Diese Rolle ermöglicht es Amazon Keyspaces, Schreibvorgänge in alle Replikate einer multiregionalen Tabelle in Ihrem Namen zu replizieren. Weitere Informationen hierzu finden Sie unter [the section called “Multiregionale Replikation”](#).

## Erstellen einer multiregionalen Tabelle mit Standardeinstellungen (Konsole)

Gehen Sie wie folgt vor, um eine multiregionale Tabelle mit der Amazon Keyspaces-Konsole zu erstellen.

So erstellen Sie eine multiregionale Tabelle (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie einen multiregionalen Keyspace aus.
3. Wählen Sie auf der Registerkarte Tabellen die Option Tabelle erstellen aus.
4. Geben Sie für Tabellennamen den Namen für die Tabelle ein. Die AWS-Regionen, in die diese Tabelle repliziert wird, werden im Informationsfeld angezeigt.
5. Fahren Sie mit dem Tabellenschema fort.
6. Fahren Sie unter Tabelleneinstellungen mit der Option Standardeinstellungen fort. Beachten Sie die folgenden Standardeinstellungen für multiregionale Tabellen.
  - Kapazitätsmodus – Der Standardkapazitätsmodus ist On-Demand. Weitere Informationen zur Konfiguration des Modus bereitgestellter Kapazität finden Sie unter [the section called “Erstellen einer multiregionalen Tabelle im Bereitstellungsmodus mit aktiviertem Auto Scaling \(Konsole\)”](#).
  - Verwaltung von Verschlüsselungsschlüsseln – Es wird nur die AWS-eigener Schlüssel Option unterstützt.
  - Clientseitige Zeitstempel – Diese Funktion ist für Tabellen mit mehreren Regionen erforderlich.
  - Wählen Sie Einstellungen anpassen, wenn Sie Time to Live (TTL) für die Tabelle und alle ihre Replikate aktivieren müssen.

### Note

Sie können die TTL-Einstellungen für eine vorhandene Tabelle mit mehreren Regionen nicht ändern.

7. Um den Vorgang abzuschließen, wählen Sie Tabelle erstellen aus.

## Erstellen einer multiregionalen Tabelle im Bereitstellungsmodus mit aktiviertem Auto Scaling (Konsole)

### Note

Die automatische Skalierung von Amazon Keyspaces erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), die automatische Skalierungsaktionen in Ihrem Namen durchführt. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

So erstellen Sie eine neue multiregionale Tabelle mit aktivierter Auto Scaling


1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie einen multiregionalen Keyspace aus.
3. Wählen Sie auf der Registerkarte Tabellen die Option Tabelle erstellen aus.
4. Wählen Sie auf der Seite Tabelle erstellen im Abschnitt Tabellendetails einen Keyspace aus und geben Sie einen Namen für die neue Tabelle ein.
5. Erstellen Sie im Abschnitt Spalten das Schema für Ihre Tabelle.
6. Definieren Sie im Abschnitt Primärschlüssel den Primärschlüssel der Tabelle und wählen Sie optionale Clustering-Spalten aus.
7. Wählen Sie im Abschnitt Tabelleneinstellungen die Option Einstellungen anpassen aus.
8. Fahren Sie mit den Lese-/Schreibkapazitätseinstellungen fort.
9. Wählen Sie für Kapazitätsmodus die Option Bereitgestellt aus.
10. Vergewissern Sie sich im Abschnitt Lesekapazität, dass Automatische Skalierung ausgewählt ist.

Sie können auswählen, ob dieselben Lesekapazitätseinheiten für alle AWS-Regionen konfiguriert werden sollen, in denen die Tabelle repliziert wird. Alternativ können Sie das Kontrollkästchen deaktivieren und die Lesekapazität für jede Region unterschiedlich konfigurieren.


Wenn Sie jede Region unterschiedlich konfigurieren, wählen Sie die minimalen und maximalen Lesekapazitätseinheiten für jedes Tabellenreplikat sowie die Zielauslastung aus.



- **Minimale Kapazitätseinheiten** – Geben Sie den Wert für den Minstdurchsatz ein, den die Tabelle immer unterstützen soll. Der Wert muss zwischen 1 und dem maximalen Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
- **Maximale Kapazitätseinheiten** – Geben Sie die maximale Durchsatzmenge ein, die Sie für die Tabelle bereitstellen möchten. Der Wert muss zwischen 1 und dem maximalen Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
- **Zielauslastung** – Geben Sie eine Zielauslastungsrate zwischen 20 % und 90 % ein. Wenn der Datenverkehr die definierte Zielauslastungsrate überschreitet, wird die Kapazität automatisch hochskaliert. Wenn der Datenverkehr unter das definierte Ziel fällt, wird er automatisch wieder herunterskaliert.
- Deaktivieren Sie das Kontrollkästchen Automatische Skalierung, wenn Sie die Lesekapazität der Tabelle manuell bereitstellen möchten. Diese Einstellung gilt für alle Replikate der Tabelle.

 Note

Um sicherzustellen, dass für alle Replikate genügend Lesekapazität vorhanden ist, empfehlen wir die automatische Skalierung von Amazon Keyspaces für bereitgestellte multiregionale Tabellen.

 Note

Weitere Informationen zu Standardkontingenten für Ihr Konto und wie Sie diese erhöhen können, finden Sie unter [Kontingente](#).

11. Vergewissern Sie sich im Abschnitt Schreibkapazität, dass Automatische Skalierung ausgewählt ist. Konfigurieren Sie dann die Kapazitätseinheiten für die Tabelle. Die Schreibkapazitätseinheiten bleiben über alle hinweg synchronisiert, AWS-Regionen um sicherzustellen, dass genügend Kapazität vorhanden ist, um Schreibereignisse in den Regionen zu replizieren.
  - Deaktivieren Sie Automatische Skalierung, wenn Sie die Schreibkapazität der Tabelle manuell bereitstellen möchten. Diese Einstellung gilt für alle Replikate der Tabelle.

**Note**

Um sicherzustellen, dass für alle Replikate genügend Schreibkapazität vorhanden ist, empfehlen wir die automatische Skalierung von Amazon Keyspaces für bereitgestellte multiregionale Tabellen.

12. Wählen Sie **Create table** (Tabelle erstellen) aus. Ihre Tabelle wird mit den angegebenen Auto-Scaling-Parametern erstellt.

## Aktivieren der automatischen Skalierung für eine vorhandene multiregionale Tabelle (Konsole)

Gehen Sie wie folgt vor, um die automatische Skalierung für eine Tabelle mit mehreren Regionen im Bereitstellungsmodus mit der Amazon Keyspaces-Konsole zu aktivieren.

**Note**

Die automatische Skalierung von Amazon Keyspaces erfordert das Vorhandensein einer serviceverknüpften Rolle (`AWSServiceRoleForApplicationAutoScaling_CassandraTable`), die automatische Skalierungsaktionen in Ihrem Namen durchführt. Diese Rolle wird automatisch für Sie erstellt. Weitere Informationen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

So aktivieren Sie die automatische Skalierung von Amazon Keyspaces für eine vorhandene Tabelle mit mehreren Regionen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie die Tabelle aus, mit der Sie arbeiten möchten, und wechseln Sie zur Registerkarte **Kapazität**.
3. Wählen Sie im Abschnitt **Kapazitätseinstellungen** die Option **Bearbeiten** aus.
4. Stellen Sie unter **Kapazitätsmodus** sicher, dass die Tabelle den Modus **Bereitgestellte Kapazität** verwendet.

5. Wählen Sie Automatisch skalieren und sehen Sie sich Schritt 9 in an, [Erstellen einer multiregionalen Tabelle im Bereitstellungsmodus mit aktiviertem Auto Scaling \(Konsole\)](#) um die Lese- und Schreibkapazität zu bearbeiten.
6. Wenn die Einstellungen für die automatische Skalierung definiert sind, wählen Sie Speichern aus.

## Deaktivieren der automatischen Skalierung für eine Tabelle mit mehreren Regionen (Konsole)

Gehen Sie wie folgt vor, um die automatische Skalierung für eine Tabelle mit mehreren Regionen im Bereitstellungsmodus mit der Amazon Keyspaces-Konsole zu deaktivieren.

So deaktivieren Sie die automatische Skalierung von Amazon Keyspaces für eine vorhandene Tabelle mit mehreren Regionen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie die Tabelle aus, mit der Sie arbeiten möchten, und wählen Sie die Registerkarte Kapazität.
3. Wählen Sie im Abschnitt Kapazitätseinstellungen die Option Bearbeiten aus.
4. Um die automatische Skalierung von Amazon Keyspaces zu deaktivieren, deaktivieren Sie das Kontrollkästchen Automatische Skalierung. Wenn Sie die automatische Skalierung deaktivieren, wird die Tabelle mit Application Auto Scaling als skalierbares Ziel abgemeldet. Um die serviceverknüpfte Rolle zu löschen, die Application Auto Scaling für den Zugriff auf Ihre Amazon Keyspaces-Tabelle verwendet, führen Sie die Schritte unter [the section called "Löschen einer serviceverknüpften Rolle für Amazon Keyspaces"](#).

### Note

Um die serviceverknüpfte Rolle zu löschen, die Application Auto Scaling verwendet, müssen Sie die automatische Skalierung für alle Tabellen im Konto in allen deaktivieren AWS-Regionen.

5. Wenn die Einstellungen für die automatische Skalierung definiert sind, wählen Sie Speichern aus.

## Anzeigen von Auto-Scaling-Aktivitäten von Amazon Keyspaces in der Konsole

Sie können überwachen, wie Amazon Keyspaces Auto Scaling Ressourcen nutzt, indem Sie Amazon verwenden CloudWatch, das Metriken über Ihre Nutzung und Leistung generiert. Führen Sie die Schritte im [Application Auto Scaling -Benutzerhandbuch aus](#), um ein CloudWatch Dashboard zu erstellen.

## Verwenden von CQL zum Erstellen und Verwalten von multiregionalen Tabellen

Sie können Cassandra Query Language (CQL) verwenden, um multiregionale Keyspaces und Tabellen in Amazon Keyspaces zu erstellen und zu verwalten.

Dieser Abschnitt enthält Beispiele für das Erstellen und Verwalten von multiregionalen Tabellen mit CQL. Alle Tabellen, die Sie in einem multiregionalen Keyspace erstellen, erben automatisch die multiregionalen Einstellungen vom Keyspace. Weitere Informationen zu CQL finden Sie in der [Amazon Keyspaces CQL-Sprachreferenz](#).

Weitere Informationen zu unterstützten Konfigurationen und Funktionen finden Sie unter [the section called "Nutzungshinweise"](#).

### Themen

- [Erstellen eines multiregionalen Keyspace \(CQL\)](#)
- [Erstellen einer multiregionalen Tabelle mit Standardeinstellungen \(CQL\)](#)
- [Erstellen einer multiregionalen Tabelle mit bereitgestelltem Kapazitätsmodus und Auto Scaling \(CQL\)](#)
- [Aktualisieren der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen \(CQL\)](#)
- [Anzeigen der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen \(CQL\)](#)
- [Deaktivieren der automatischen Skalierung für eine Tabelle mit mehreren Regionen \(CQL\)](#)
- [Manuelles Festlegen der bereitgestellten Kapazität einer Tabelle mit mehreren Regionen \(CQL\)](#)

### Erstellen eines multiregionalen Keyspace (CQL)

Um einen multiregionalen Keyspace zu erstellen, verwenden Sie `NetworkTopologyStrategy` um die anzugeben AWS-Regionen, in die der Keyspace repliziert werden soll. Sie müssen Ihre

aktuelle Region und mindestens eine zusätzliche Region angeben. Die folgende CQL-Anweisung ist ein Beispiel dafür.

```
CREATE KEYSPACE mykeyspace
WITH REPLICATION = {'class': 'NetworkTopologyStrategy', 'us-east-1': '3', 'ap-southeast-1': '3', 'eu-west-1': '3' };
```

Alle Tabellen im Keyspace verwenden dieselbe Replikationsstrategie wie der Keyspace. Sie können die Replikationsstrategie auf Tabellenebene nicht ändern.

NetworkTopologyStrategy – Der Replikationsfaktor für jede Region beträgt drei AWS-Region, da Amazon Keyspaces Daten standardmäßig über drei [Availability Zones](#) innerhalb derselben repliziert.

### Note

Beim Erstellen eines multiregionalen Keyspace erstellt Amazon Keyspaces eine serviceverknüpfte Rolle mit dem Namen `AWSServiceRoleForAmazonKeyspacesReplication` in Ihrem Konto. Diese Rolle ermöglicht es Amazon Keyspaces, Schreibvorgänge in alle Replikate einer multiregionalen Tabelle in Ihrem Namen zu replizieren. Weitere Informationen hierzu finden Sie unter [the section called “Multiregionale Replikation”](#).

Sie können eine CQL-Anweisung verwenden, um die `tables` Tabelle im `system_multiregion_info` Keyspace abzufragen und die Regionen und den Status der von Ihnen angegebenen Tabelle mit mehreren Regionen programmgesteuert aufzulisten. Der folgende Code ist ein Beispiel dafür.

```
SELECT * from system_multiregion_info.tables WHERE keyspace_name = 'mykeyspace' AND
table_name = 'mytable';
```

Die Ausgabe der Anweisung sieht wie folgt aus:

| keyspace_name | table_name | region         | status |
|---------------|------------|----------------|--------|
| mykeyspace    | mytable    | us-east-1      | ACTIVE |
| mykeyspace    | mytable    | ap-southeast-1 | ACTIVE |
| mykeyspace    | mytable    | eu-west-1      | ACTIVE |

## Erstellen einer multiregionalen Tabelle mit Standardeinstellungen (CQL)

Verwenden Sie das folgende Beispiel, um eine multiregionale Tabelle mit Standardeinstellungen zu erstellen.

```
CREATE TABLE mykeyspace.mytable(pk int, ck int, PRIMARY KEY (pk, ck))
 WITH CUSTOM_PROPERTIES = {
 'capacity_mode':{
 'throughput_mode':'PAY_PER_REQUEST'
 },
 'point_in_time_recovery':{
 'status':'enabled'
 },
 'encryption_specification':{
 'encryption_type':'AWS_OWNED_KMS_KEY'
 },
 'client_side_timestamps':{
 'status':'enabled'
 }
};
```

## Erstellen einer multiregionalen Tabelle mit bereitgestelltem Kapazitätsmodus und Auto Scaling (CQL)

Um eine multiregionale Tabelle im Bereitstellungsmodus mit Auto Scaling zu erstellen, müssen Sie zuerst den Kapazitätsmodus angeben, indem Sie CUSTOM\_PROPERTIES für die Tabelle definieren. Nachdem Sie den Modus bereitgestellter Kapazität angegeben haben, können Sie die Auto-Scaling-Einstellungen für die Tabelle mit konfigurierenAUTOSCALING\_SETTINGS.

Ausführliche Informationen zu Auto-Scaling-Einstellungen, der Richtlinie für die Ziel-Nachverfolgung, dem Zielwert und optionalen Einstellungen finden Sie unter [the section called “Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe von CQL”](#).

Wenn Sie eine multiregionale Tabelle erstellen, können Sie für jedes Replikat der Tabelle auch unterschiedliche Lesekapazitäts- und Lese-Auto-Scaling-Einstellungen angeben. Die von Ihnen angegebenen Einstellungen überschreiben die allgemeinen Einstellungen der Tabelle für das angegebene AWS-Region. Die Schreibkapazität bleibt jedoch zwischen allen Replikaten synchronisiert, um sicherzustellen, dass genügend Kapazität vorhanden ist, um Schreibvorgänge in allen Regionen zu replizieren.

Um die Lesekapazität für ein Tabellenreplikate in einer bestimmten Region zu definieren, können Sie die folgenden Parameter als Teil der Tabelle konfigurieren `replica_updates`:

- Die Region
- Die bereitgestellten Lesekapazitätseinheiten (optional)
- Auto-Scaling-Einstellungen für die Lesekapazität (optional)

Das folgende Beispiel zeigt eine `-CREATE TABLE`Anweisung für eine Tabelle mit mehreren Regionen im Bereitstellungsmodus. Die allgemeinen Einstellungen für die automatische Skalierung der Schreib- und Lesekapazität sind identisch. Die Einstellungen für die automatische Skalierung von Lesevorgängen legen jedoch zusätzliche Ruhephasen von 60 Sekunden fest, bevor die Lesekapazität der Tabelle nach oben oder unten skaliert wird. Darüber hinaus sind die Einstellungen für die automatische Skalierung der Lesekapazität für die Region USA Ost (Nord-Virginia) höher als für andere Replikate. Außerdem wird der Zielwert auf 70 % statt auf 50 % festgelegt.

```
CREATE TABLE mykeyspace.mytable(pk int, ck int, PRIMARY KEY (pk, ck))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 5,
 'write_capacity_units': 5
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 }
}
```

```

 }
 }
},
'replica_updates': {
 'us-east-1': {
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 20,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 70
 }
 }
 }
 }
}
};

```

## Aktualisieren der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen (CQL)

Sie können verwenden `ALTER TABLE`, um den Kapazitätsmodus und die Auto-Scaling-Einstellungen einer vorhandenen Tabelle zu aktualisieren. Wenn Sie eine Tabelle aktualisieren, die sich derzeit im On-Demand-Kapazitätsmodus befindet, `capacity_mode` ist erforderlich. Wenn sich Ihre Tabelle bereits im Modus bereitgestellter Kapazität befindet, kann dieses Feld weggelassen werden.

Ausführliche Informationen zu Auto-Scaling-Einstellungen, der Richtlinie für die Ziel-Nachverfolgung, dem Zielwert und optionalen Einstellungen finden Sie unter [the section called “Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe von CQL”](#).

In derselben Anweisung können Sie auch die Lesekapazität und die Auto-Scaling-Einstellungen von Tabellenreplikaten in bestimmten Regionen aktualisieren, indem Sie die `-replica_updates`Eigenschaft der Tabelle aktualisieren. Die folgende Anweisung ist ein Beispiel dafür.

```

ALTER TABLE mykeyspace.mytable
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 }
};

```



```

 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 },
 'replica_updates': {
 'us-east-1': {
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 20,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 70
 }
 }
 }
 }
 }
};

```

## Anzeigen der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen (CQL)

Verwenden Sie den folgenden Befehl, um die Auto-Scaling-Konfiguration einer Tabelle mit mehreren Regionen anzuzeigen.

```
SELECT * FROM system_multiregion_info.autoscaling WHERE keyspace_name = 'mykeyspace'
AND table_name = 'mytable';
```

Die Ausgabe für diesen Befehl sieht wie folgt aus:

```
keyspace_name | table_name | region |
provisioned_read_capacity_autoscaling_update
 | provisioned_write_capacity_autoscaling_update
-----+-----+-----
+-----+-----+-----
mykeyspace | mytable | ap-southeast-1 | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 60, 'disable_scale_in': false, 'target_value':
50, 'scale_in_cooldown': 60}}} | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 0, 'disable_scale_in': false, 'target_value': 50,
'scale_in_cooldown': 0}}}
mykeyspace | mytable | us-east-1 | {'minimum_units': 5, 'maximum_units':
20, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 60, 'disable_scale_in': false, 'target_value':
70, 'scale_in_cooldown': 60}}} | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 0, 'disable_scale_in': false, 'target_value': 50,
'scale_in_cooldown': 0}}}
mykeyspace | mytable | eu-west-1 | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 60, 'disable_scale_in': false, 'target_value':
50, 'scale_in_cooldown': 60}}} | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 0, 'disable_scale_in': false, 'target_value': 50,
'scale_in_cooldown': 0}}}
```

## Deaktivieren der automatischen Skalierung für eine Tabelle mit mehreren Regionen (CQL)

Sie können verwenden `ALTER TABLE`, um die automatische Skalierung für eine vorhandene Tabelle zu deaktivieren. Beachten Sie, dass Sie die automatische Skalierung für ein einzelnes Tabellenreplikat nicht deaktivieren können.

Im folgenden Beispiel ist die automatische Skalierung für die Lesekapazität der Tabelle deaktiviert.

```
ALTER TABLE mykeyspace.mytable
WITH AUTOSCALING_SETTINGS = {
 'provisioned_read_capacity_autoscaling_update': {
 'autoscaling_disabled': true
 }
};
```

### Note

Um die von Application Auto Scaling verwendete serviceverknüpfte Rolle zu löschen, müssen Sie die automatische Skalierung für alle Tabellen im Konto für alle deaktivieren AWS-Regionen.

## Manuelles Festlegen der bereitgestellten Kapazität einer Tabelle mit mehreren Regionen (CQL)

Wenn Sie Auto Scaling für eine Tabelle mit mehreren Regionen deaktivieren müssen, können Sie verwenden, `ALTER TABLE` um die Lesekapazität der Tabelle für eine Replikattabelle manuell bereitzustellen.

```
ALTER TABLE mykeyspace.mytable
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 },
 'replica_updates': {
 'us-east-1': {
 'read_capacity_units': 2
 }
 }
};
```

**Note**

Wir empfehlen die Verwendung von Auto Scaling für Tabellen mit mehreren Regionen, die bereitgestellte Kapazität verwenden. Weitere Informationen finden Sie unter [the section called “Multiregionale Tabellen”](#).

## Verwenden der AWS CLI zum Erstellen und Verwalten von multiregionalen Tabellen

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um multiregionale Keyspaces und Tabellen in Amazon Keyspaces zu erstellen und zu verwalten.

Dieser Abschnitt enthält Beispiele für das Erstellen und Verwalten von multiregionalen Tabellen mit der AWS CLI. Alle Tabellen, die Sie in einem multiregionalen Keyspace erstellen, erben automatisch die multiregionalen Einstellungen vom Keyspace.

Weitere Informationen zu den in diesem Thema beschriebenen Amazon Keyspaces- AWS CLI Befehlen finden Sie in der [-AWS CLI Befehlsreferenz für Amazon Keyspaces](#).

### Themen

- [Erstellen eines neuen multiregionalen Keyspace \(CLI\)](#)
- [Erstellen einer neuen multiregionalen Tabelle mit Standardeinstellungen \(CLI\)](#)
- [Erstellen einer neuen multiregionalen Tabelle im Bereitstellungsmodus mit Auto Scaling \(CLI\)](#)
- [Aktualisieren der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen \(CLI\)](#)
- [Anzeigen der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen \(CLI\)](#)
- [Deaktivieren der automatischen Skalierung für eine Tabelle mit mehreren Regionen \(CLI\)](#)
- [Manuelles Festlegen der bereitgestellten Kapazität einer Tabelle mit mehreren Regionen \(CLI\)](#)

### Erstellen eines neuen multiregionalen Keyspace (CLI)

Um einen multiregionalen Keyspace zu erstellen, können Sie die folgende CLI-Anweisung verwenden. Geben Sie Ihre aktuelle Region und mindestens eine zusätzliche Region in der `anregionList`.

```
aws keyspaces create-keyspace --keyspace-name mykeyspace
 \ --replication-specification
 replicationStrategy=MULTI_REGION,regionList=us-east-1,eu-west-1
```

### Note

Beim Erstellen eines multiregionalen Keyspace erstellt Amazon Keyspaces eine serviceverknüpfte Rolle mit dem Namen `AWSServiceRoleForAmazonKeyspacesReplication` in Ihrem Konto. Diese Rolle ermöglicht es Amazon Keyspaces, Schreibvorgänge in alle Replikate einer multiregionalen Tabelle in Ihrem Namen zu replizieren. Weitere Informationen hierzu finden Sie unter [the section called “Multiregionale Replikation”](#).

## Erstellen einer neuen multiregionalen Tabelle mit Standardeinstellungen (CLI)

Um eine multiregionale Tabelle mit Standardeinstellungen zu erstellen, müssen Sie nur das Schema angeben. Sie können das folgende Beispiel verwenden.

```
aws keyspaces create-table --keyspace-name mykeyspace --table-name mytable
 \ --schema-definition 'allColumns=[{name=pk,type=int}],partitionKeys={name=
 pk}'
```

Die Ausgabe des Befehls lautet:

```
{
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
 table/mytable"
}
```

Um die Einstellungen der Tabelle zu bestätigen, können Sie die folgende Anweisung verwenden.

```
aws keyspaces get-table --keyspace-name mykeyspace --table-name mytable
```

Die Ausgabe zeigt alle Standardeinstellungen einer Tabelle mit mehreren Regionen.

```
{
 "keyspaceName": "mykeyspace",
 "tableName": "mytable",
```

```
"resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/mytable",
"creationTimestamp": "2023-12-19T16:50:37.639000+00:00",
"status": "ACTIVE",
"schemaDefinition": {
 "allColumns": [
 {
 "name": "pk",
 "type": "int"
 }
],
 "partitionKeys": [
 {
 "name": "pk"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
},
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2023-12-19T16:50:37.639000+00:00"
},
"encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
 "status": "DISABLED"
},
"defaultTimeToLive": 0,
"comment": {
 "message": ""
},
"clientSideTimestamps": {
 "status": "ENABLED"
},
"replicaSpecifications": [
 {
 "region": "us-east-1",
 "status": "ACTIVE",
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1702895811.469
 }
 }
]
```

```
 },
 {
 "region": "eu-north-1",
 "status": "ACTIVE",
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1702895811.121
 }
 }
]
}
```

## Erstellen einer neuen multiregionalen Tabelle im Bereitstellungsmodus mit Auto Scaling (CLI)

Um eine multiregionale Tabelle im Bereitstellungsmodus mit Auto-Scaling-Konfiguration zu erstellen, können Sie die verwenden AWS CLI. Beachten Sie, dass Sie den Amazon Keyspaces CLI-`create-table`-Befehl verwenden müssen, um Einstellungen für die automatische Skalierung in mehreren Regionen zu konfigurieren. Dies liegt daran, dass Application Auto Scaling, der Service, den Amazon Keyspaces zur Durchführung der automatischen Skalierung in Ihrem Namen verwendet, nicht mehrere Regionen unterstützt.

Weitere Informationen zu Auto-Scaling-Einstellungen, der Zielverfolgungsrichtlinie, dem Zielwert und optionalen Einstellungen finden Sie unter [the section called “Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe der AWS CLI”](#).

Wenn Sie eine neue multiregionale Tabelle im Bereitstellungsmodus mit Auto-Scaling-Einstellungen erstellen, können Sie die allgemeinen Einstellungen für die Tabelle angeben, die für alle gültig sind AWS-Regionen, in denen die Tabelle repliziert wird. Anschließend können Sie die Einstellungen für die Lesekapazität und die automatische Skalierung von Lesevorgängen für jedes Replikat überschreiben. Die Schreibkapazität bleibt jedoch zwischen allen Replikaten synchronisiert, um sicherzustellen, dass genügend Kapazität vorhanden ist, um Schreibvorgänge in allen Regionen zu replizieren.

Um die Lesekapazität für ein Tabellenreplikat in einer bestimmten Region zu definieren, können Sie die folgenden Parameter als Teil der Tabelle konfigurieren `replicaSpecifications`:

- Die Region
- Die bereitgestellten Lesekapazitätseinheiten (optional)
- Auto-Scaling-Einstellungen für die Lesekapazität (optional)

Wenn Sie bereitgestellte multiregionale Tabellen mit komplexen Auto-Scaling-Einstellungen und unterschiedlichen Konfigurationen für Tabellenreplikate erstellen, ist es hilfreich, die Auto-Scaling-Einstellungen und Replikatkonfigurationen der Tabelle aus JSON-Dateien zu laden.

Um das folgende Codebeispiel zu verwenden, können Sie die Beispiel-JSON-Dateien von [auto-scaling.zip](#) herunterladen und `auto-scaling.json` und `extrahierenreplication.json` extrahieren. Notieren Sie sich den Pfad zu den Dateien.

In diesem Beispiel befinden sich die JSON-Dateien im aktuellen Verzeichnis. Informationen zu verschiedenen Dateipfadoptionen finden Sie unter [So laden Sie Parameter aus einer Datei](#).

```
aws keyspaces create-table --keyspace-name mykeyspace --table-name mytable
 \ --schema-definition 'allColumns=[{name=pk,type=int},
{name=ck,type=int}],partitionKeys=[{name=pk},{name=ck}]'
 \ --capacity-specification
throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --auto-scaling-specification file://auto-scaling.json
 \ --replica-specifications file://replication.json
```

## Aktualisieren der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen (CLI)

Um den Bereitstellungsmodus und die Auto-Scaling-Konfiguration einer vorhandenen Tabelle zu aktualisieren, können Sie den AWS CLI `update-table` Befehl verwenden.

Beachten Sie, dass Sie die Amazon-Keyspaces-CLI-Befehle verwenden müssen, um multiregionale Auto-Scaling-Einstellungen zu erstellen oder zu ändern. Dies liegt daran, dass Application Auto Scaling, der Service, den Amazon Keyspaces zur automatischen Skalierung der Tabellenkapazität in Ihrem Namen verwendet, nicht mehrere unterstützt AWS-Regionen.

Wenn Sie den bereitgestellten Modus oder die Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen aktualisieren, können Sie die Einstellungen für die Lesekapazität und die Konfiguration für die automatische Leseskalierung für jedes Replikat der Tabelle aktualisieren.

Die Schreibkapazität bleibt jedoch zwischen allen Replikaten synchronisiert, um sicherzustellen, dass genügend Kapazität vorhanden ist, um Schreibvorgänge in allen Regionen zu replizieren. Um die Lesekapazität für ein Tabellenreplikat in einer bestimmten Region zu aktualisieren, können Sie einen der folgenden optionalen Parameter des der Tabelle `ändernreplicaSpecifications`:

- Die bereitgestellten Lesekapazitätseinheiten (optional)



- Auto-Scaling-Einstellungen für die Lesekapazität (optional)

Wenn Sie multiregionale Tabellen mit komplexen Auto-Scaling-Einstellungen und unterschiedlichen Konfigurationen für Tabellenreplikate aktualisieren, ist es hilfreich, die Auto-Scaling-Einstellungen und Replikatkonfigurationen der Tabelle aus JSON-Dateien zu laden.

Um das folgende Codebeispiel zu verwenden, können Sie die Beispiel-JSON-Dateien von [auto-scaling.zip](#) herunterladen und `auto-scaling.json` und `replication.json` extrahieren. Notieren Sie sich den Pfad zu den Dateien.

In diesem Beispiel befinden sich die JSON-Dateien im aktuellen Verzeichnis. Informationen zu verschiedenen Dateipfadoptionen finden Sie unter [So laden Sie Parameter aus einer Datei](#).

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --auto-scaling-specification file://auto-scaling.json
 \ --replica-specifications file://replication.json
```

## Anzeigen der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen (CLI)

Um die Auto-Scaling-Konfiguration einer Tabelle mit mehreren Regionen anzuzeigen, können Sie die `-get-table-auto-scaling-settings` Operation verwenden. Der folgende CLI-Befehl ist ein Beispiel dafür.

```
aws keyspaces get-table-auto-scaling-settings --keyspace-name mykeyspace --table-name
mytable
```

Die Ausgabe sollte folgendermaßen aussehen.

```
{
 "keyspaceName": "mykeyspace",
 "tableName": "mytable",
 "resourceArn": "arn:aws:cassandra:us-east-1:777788889999:/keyspace/mykeyspace/
table/mytable",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
```

```
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 20,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 70.0
 }
 }
 }
},
"replicaSpecifications": [
 {
 "region": "us-east-1",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
```

```
 "maximumUnits": 20,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 70.0
 }
 }
 },
},
{
 "region": "eu-north-1",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 50.0
 }
 }
 }
 }
}
]
```

```
}
```

## Deaktivieren der automatischen Skalierung für eine Tabelle mit mehreren Regionen (CLI)

Sie können den AWS CLI `update-table` Befehl verwenden, um die automatische Skalierung für eine vorhandene Tabelle zu deaktivieren. Beachten Sie, dass Sie die automatische Skalierung für ein einzelnes Tabellenreplikat nicht deaktivieren können.

Im folgenden Beispiel ist die automatische Skalierung für die Lesekapazität der Tabelle deaktiviert.

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --auto-scaling-specification
 readCapacityAutoScaling={autoScalingDisabled=true}
```


### Note

Um die serviceverknüpfte Rolle zu löschen, die von Application Auto Scaling verwendet wird, müssen Sie die automatische Skalierung für alle Tabellen im Konto für alle deaktivieren AWS-Regionen.

## Manuelles Festlegen der bereitgestellten Kapazität einer Tabelle mit mehreren Regionen (CLI)

Wenn Sie die automatische Skalierung für eine Tabelle mit mehreren Regionen deaktivieren müssen, können Sie verwenden, `update-table` um die Lesekapazität der Tabelle für eine Replikattabelle manuell bereitzustellen.

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --replica-specifications region="us-east-1",readCapacityUnits=5
```

 Note

Wir empfehlen die Verwendung von Auto Scaling für multiregionale Tabellen, die bereitgestellte Kapazität verwenden. Weitere Informationen finden Sie unter [the section called “Multiregionale Tabellen”](#).

# Point-in-time P-Wiederherstellung für Amazon Keyspaces (für Apache Cassandra)

Point-in-time Recovery (PITR) schützt Ihre Amazon Keyspaces-Tabellen vor versehentlichen Schreib- oder Löschvorgängen, indem es Ihnen kontinuierliche Backups Ihrer Tabellendaten bereitstellt.

Angenommen, Sie schreiben einen Skript-Test versehentlich in eine Amazon Keyspaces-Produktionstabelle. Mit der point-in-time Wiederherstellung können Sie die Daten dieser Tabelle im Zustand eines beliebigen Zeitpunkts seit der Aktivierung von PITR innerhalb der vergangenen 35 Tage wiederherstellen. Wenn Sie eine Tabelle mit aktivierter point-in-time Wiederherstellung löschen, erstellen Sie 35 Tage ein Backup, das 35 Tage aufbewahrt wird (ohne zusätzliche Kosten), um den Zustand wiederherzustellen, in dem sie sich kurz vor dem Löschen befanden.

Sie können eine Amazon Keyspaces-Tabelle auf einen bestimmten Zeitpunkt wiederherstellen, indem Sie die Konsole, das AWS SDK und die AWS Command Line Interface (AWS CLI) oder Cassandra Query Language (CQL) verwenden. Weitere Informationen finden Sie unter [Wiederherstellen einer Amazon Keyspaces-Tabelle auf einen bestimmten Zeitpunkt](#).

Point-in-time P-Operationen haben keine Auswirkungen auf die Leistung oder Verfügbarkeit der Basistabelle, und das Wiederherstellen einer Tabelle verbraucht keinen zusätzlichen Durchsatz.

Hinweise zu den PITR-Kontingenten finden Sie unter [Kontingente](#).

Hinweise zur Preisgestaltung finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\) -Preise](#).

Themen

- [Funktionsweise der point-in-time Wiederherstellung in Amazon Keyspaces](#)
- [Wiederherstellen einer Amazon Keyspaces-Tabelle auf einen bestimmten Zeitpunkt](#)

## Funktionsweise der point-in-time Wiederherstellung in Amazon Keyspaces

Dieser Abschnitt bietet einen Überblick über die Funktionsweise der Amazon Keyspaces- point-in-time Wiederherstellung (PITR). Weitere Informationen zu Preisen finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\) – Preise](#).

Themen

- [Aktivieren der point-in-time Wiederherstellung \(PITR\)](#)
- [Erforderliche Berechtigungen zum Wiederherstellen einer Tabelle](#)
- [Zeitfenster für kontinuierliche PITR-Backups](#)
- [PITR-Wiederherstellungseinstellungen](#)
- [PITR-Wiederherstellung verschlüsselter Tabellen](#)
- [PITR-Wiederherstellung von multiregionalen Tabellen](#)
- [Tabellenwiederherstellungszeit mit PITR](#)
- [Amazon Keyspaces PITR und Integration in -AWSservices](#)

## Aktivieren der point-in-time Wiederherstellung (PITR)

Sie können PITR über die Konsole oder programmgesteuert aktivieren.

### Aktivieren von PITR mit der Konsole

PITR-Einstellungen für neue Tabellen können unter der Option Benutzerdefinierte Einstellungen verwaltet werden. PITR ist standardmäßig für neue Tabellen aktiviert, die über die Konsole erstellt wurden.

Führen Sie die folgenden Schritte aus, um PITR für eine vorhandene Tabelle zu aktivieren.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tabellen und dann die Tabelle aus, die Sie bearbeiten möchten.
3. Wählen Sie auf der Registerkarte Backups die Option Bearbeiten aus.
4. Wählen Sie im Abschnitt point-in-time Wiederherstellungseinstellungen bearbeiten die Option Point-in-time Wiederherstellung aktivieren aus.

Sie können PITR für eine Tabelle jederzeit mit den folgenden Schritten deaktivieren.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tabellen und dann die Tabelle aus, die Sie bearbeiten möchten.

3. Wählen Sie auf der Registerkarte Backups die Option Bearbeiten aus.
4. Deaktivieren Sie im Abschnitt `point-in-time` Wiederherstellungseinstellungen bearbeiten das Kontrollkästchen `Point-in-time -Wiederherstellung` aktivieren.

#### Important

Wenn Sie PITR deaktivieren, wird Ihr Backup-Verlauf sofort gelöscht, auch wenn Sie PITR innerhalb von 35 Tagen erneut für die Tabelle aktivieren.

Informationen zum Wiederherstellen einer Tabelle mithilfe der Konsole finden Sie unter [the section called “Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt \(Konsole\)”](#).

## Aktivieren von PITR mithilfe der AWS CLI

Sie können PITR-Einstellungen für Tabellen mithilfe der `UpdateTable`-API verwalten.

Wenn Sie eine neue Tabelle mit erstellen AWS CLI, müssen Sie PITR explizit aktivieren, wenn Sie die neue Tabelle erstellen.

Um PITR beim Erstellen einer neuen Tabelle zu aktivieren, können Sie den folgenden AWS CLI Befehl als Beispiel verwenden. Der Befehl wurde zur besseren Lesbarkeit in separate Zeilen unterteilt.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable'
 --schema-definition 'allColumns=[{name=id,type=int},{name=name,type=text},
{name=date,type=timestamp}],partitionKeys=[{name=id}]'
 --point-in-time-recovery 'status=ENABLED'
```

#### Note

Wenn kein `point-in-time` Wiederherstellungswert angegeben ist, `point-in-time` ist die Wiederherstellung standardmäßig deaktiviert.

Um die `point-in-time` Wiederherstellungseinstellung für eine Tabelle zu bestätigen, können Sie den folgenden AWS CLI Befehl verwenden.



```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

Führen Sie den folgenden Befehl aus AWS CLI, um PITR für eine vorhandene Tabelle mit der zu aktivieren.

```
aws keyspaces update-table --keyspace-name 'myKeyspace' --table-name 'myTable' --point-in-time-recovery 'status=ENABLED'
```

Führen Sie den folgenden AWS CLI Befehl aus, um PITR für eine vorhandene Tabelle zu deaktivieren.

```
aws keyspaces update-table --keyspace-name 'myKeyspace' --table-name 'myTable' --point-in-time-recovery 'status=DISABLED'
```

### Important

Wenn Sie PITR deaktivieren, wird Ihr Backup-Verlauf sofort gelöscht, auch wenn Sie PITR innerhalb von 35 Tagen erneut für die Tabelle aktivieren.

## Aktivieren von PITR mit CQL

Sie können PITR-Einstellungen für Tabellen mithilfe der `point_in_time_recovery` benutzerdefinierten Eigenschaft verwalten.

Wenn Sie eine neue Tabelle mit CQL erstellen, müssen Sie PITR explizit aktivieren, wenn Sie die neue Tabelle erstellen.

Um PITR beim Erstellen einer neuen Tabelle zu aktivieren, können Sie den folgenden CQL-Befehl als Beispiel verwenden.

```
CREATE TABLE "my_keyspace1"."my_table1"(
 "id" int,
 "name" ascii,
 "date" timestamp,
 PRIMARY KEY("id"))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode':{'throughput_mode':'PAY_PER_REQUEST'},
 'point_in_time_recovery':{'status':'enabled'}
```

}

**Note**

Wenn keine benutzerdefinierte Eigenschaft für die point-in-time Wiederherstellung angegeben ist, point-in-time ist die Wiederherstellung standardmäßig deaktiviert.

Um PITR für eine vorhandene Tabelle mit CQL zu aktivieren, führen Sie den folgenden CQL-Befehl aus.

```
ALTER TABLE mykeyspace.mytable
WITH custom_properties = {'point_in_time_recovery': {'status': 'enabled'}}
```

Um PITR für eine vorhandene Tabelle zu deaktivieren, führen Sie den folgenden CQL-Befehl aus.

```
ALTER TABLE mykeyspace.mytable
WITH custom_properties = {'point_in_time_recovery': {'status': 'disabled'}}
```

**⚠ Important**

Wenn Sie PITR deaktivieren, wird Ihr Backup-Verlauf sofort gelöscht, auch wenn Sie PITR innerhalb von 35 Tagen erneut für die Tabelle aktivieren.

Weitere Informationen in der CQL-Sprachreferenz finden Sie unter [the section called “CREATE TABLE”](#) und [the section called “ALTER TABLE”](#). Informationen zum Wiederherstellen einer Tabelle mit CQL finden Sie unter [the section called “Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit CQL”](#).

## Erforderliche Berechtigungen zum Wiederherstellen einer Tabelle

Um eine Tabelle erfolgreich wiederherzustellen, benötigt der IAM-Benutzer oder die IAM-Rolle die folgenden Mindestberechtigungen:

- `cassandra:Restore` – Die Wiederherstellungsaktion ist erforderlich, damit die Zieltabelle wiederhergestellt werden kann.
- `cassandra:Select` – Die Select-Aktion ist erforderlich, um aus der Quelltable zu lesen.

- `cassandra:TagResource` – Die Tag-Aktion ist optional und nur erforderlich, wenn der Wiederherstellungsvorgang Tags hinzufügt.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die einem Benutzer die mindestens erforderlichen Berechtigungen zum Wiederherstellen von Tabellen im Keyspace `gewährtmykeyspace`.

```
{
 "Version":"2012-10-17",
 "Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "cassandra:Restore",
 "cassandra:Select"
],
 "Resource":[
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

Je nach anderen ausgewählten Funktionen sind möglicherweise zusätzliche Berechtigungen zum Wiederherstellen einer Tabelle erforderlich. Wenn die Quelltablette beispielsweise im Ruhezustand mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, muss Amazon Keyspaces über Berechtigungen für den Zugriff auf den vom Kunden verwalteten Schlüssel der Quelltablette verfügen, um die Tabelle erfolgreich wiederherzustellen. Weitere Informationen finden Sie unter [the section called "PITR- und verschlüsselte Tabellen"](#).

Wenn Sie IAM-Richtlinien mit [Bedingungsschlüsseln](#) verwenden, um eingehenden Datenverkehr auf bestimmte Quellen zu beschränken, müssen Sie sicherstellen, dass Amazon Keyspaces über die Berechtigung verfügt, einen Wiederherstellungsvorgang im Namen Ihres Auftraggebers durchzuführen. Sie müssen Ihrer IAM-Richtlinie einen `-aws:ViaAWSServiceBedingungsschlüssel` hinzufügen, wenn Ihre Richtlinie eingehenden Datenverkehr auf eine der folgenden Optionen beschränkt:

- VPC-Endpunkte mit `aws:SourceVpce`

- IP-Bereiche mit `aws:SourceIp`
- VPCs mit `aws:SourceVpc`

Der `-aws:ViaAWSService` Bedingungsschlüssel erlaubt den Zugriff, wenn ein AWS Service eine Anforderung unter Verwendung der Anmeldeinformationen des Auftraggebers stellt. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungsschlüssel](#) im IAM-Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die den Quelldatenverkehr auf eine bestimmte IP-Adresse beschränkt und es Amazon Keyspaces ermöglicht, eine Tabelle im Namen des Auftraggebers wiederherzustellen.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CassandraAccessForCustomIp",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "false"
 },
 "ForAnyValue:IpAddress": {
 "aws:SourceIp": [
 "123.45.167.89"
]
 }
 }
 },
 {
 "Sid": "CassandraAccessForAwsService",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "true"
 }
 }
 }
]
}
```

```
 }
]
}
```

Eine Beispielrichtlinie, die den `aws:ViaAWSService` globalen Bedingungsschlüssel verwendet, finden Sie unter [the section called “VPC-Endpunktrichtlinien und Amazon point-in-time Keyspaces-Wiederherstellung \(PITR\)”](#).

## Zeitfenster für kontinuierliche PITR-Backups

Amazon Keyspaces PITR verwendet zwei Zeitstempel, um den Zeitraum beizubehalten, für den wiederherstellbare Backups für eine Tabelle verfügbar sind.

- **Früheste wiederherstellbare Zeit** – Kennzeichnet den Zeitpunkt der frühesten wiederherstellbaren Sicherung. Die früheste wiederherstellbare Sicherung wird bis zu 35 Tage oder als PITR aktiviert wurde, je nachdem, welcher Zeitpunkt neuer ist, fortgesetzt. Das maximale Backup-Fenster von 35 Tagen kann nicht geändert werden.
- **Aktuelle Zeit** – Der Zeitstempel für die letzte wiederherstellbare Sicherung ist die aktuelle Zeit. Wenn während einer Wiederherstellung kein Zeitstempel angegeben wird, wird die aktuelle Uhrzeit verwendet.

Wenn PITR aktiviert ist, können Sie zu einem beliebigen Zeitpunkt zwischen `EarliestRestorableDateTime` und `wiederherstellenCurrentTime`. Sie können Tabellendaten nur zu einem Zeitpunkt wiederherstellen, zu dem PITR aktiviert wurde.

Wenn Sie PITR deaktivieren und später wieder aktivieren, setzen Sie die Startzeit für die erste verfügbare Sicherung auf zurück, als PITR wieder aktiviert wurde. Das bedeutet, dass das Deaktivieren von PITR Ihren Backup-Verlauf löscht.

### Note

Data Definition Language (DDL)-Operationen für Tabellen, z. B. Schemaänderungen, werden asynchron ausgeführt. Sie können abgeschlossene Operationen nur in Ihren wiederhergestellten Tabellendaten sehen, aber möglicherweise werden zusätzliche Aktionen in Ihrer Quelltable angezeigt, wenn sie zum Zeitpunkt der Wiederherstellung ausgeführt wurden. Eine Liste der DDL-Anweisungen finden Sie unter [the section called “DDL-Anweisungen”](#).

Eine Tabelle muss nicht aktiv sein, um wiederhergestellt werden zu können. Sie können gelöschte Tabellen auch wiederherstellen, wenn PITR für die gelöschte Tabelle aktiviert war und das Löschen innerhalb des Backup-Fensters (oder innerhalb der letzten 35 Tage) erfolgt ist.

#### Note

Wenn eine neue Tabelle mit demselben qualifizierten Namen (z. B. `mykeyspace.mytable`) wie eine zuvor gelöschte Tabelle erstellt wird, kann die gelöschte Tabelle nicht mehr wiederhergestellt werden. Wenn Sie versuchen, dies über die Konsole zu tun, wird eine Warnung angezeigt.

## PITR-Wiederherstellungseinstellungen

Wenn Sie eine Tabelle mit PITR wiederherstellen, stellt Amazon Keyspaces das Schema und die Daten Ihrer Quelltable basierend auf dem ausgewählten Zeitstempel (`day:hour:minute:second`) in einer neuen Tabelle wieder her. PITR überschreibt keine vorhandenen Tabellen.

Zusätzlich zum Schema und den Daten der Tabelle stellt PITR die `custom_properties` aus der Quelltable wieder her. Im Gegensatz zu den Daten der Tabelle, die auf der Grundlage des ausgewählten Zeitstempels zwischen der frühesten Wiederherstellungszeit und der aktuellen Zeit wiederhergestellt werden, werden benutzerdefinierte Eigenschaften immer auf der Grundlage der Tabelleneinstellungen zum aktuellen Zeitpunkt wiederhergestellt.

Die Einstellungen der wiederhergestellten Tabelle stimmen mit den Einstellungen der Quelltable mit dem Zeitstempel überein, zu dem die Wiederherstellung initiiert wurde. Wenn Sie diese Einstellungen während der Wiederherstellung überschreiben möchten, können Sie dies mit `tunWITH custom_properties`. Zu den benutzerdefinierten Eigenschaften gehören die folgenden Einstellungen.

- Lese-/Schreibkapazitätsmodus
- Kapazitätseinstellungen für bereitgestellten Durchsatz
- PITR-Einstellungen

Wenn sich die Tabelle im Modus bereitgestellter Kapazität mit aktiviertem Auto Scaling befindet, stellt der Wiederherstellungsvorgang auch die Auto Scaling-Einstellungen der Tabelle

wieder her. Sie können sie mit dem `autoscaling_settings` Parameter in CQL oder `autoScalingSpecification` mit der CLI überschreiben. Weitere Informationen zu den Auto-Scaling-Einstellungen finden Sie unter [the section called “Verwaltung der Durchsatzkapazität mit Auto Scaling”](#).

Wenn Sie eine vollständige Tabellenwiederherstellung durchführen, stammen alle Tabelleneinstellungen für die wiederhergestellte Tabelle aus den aktuellen Einstellungen der Quelltable zu dem Zeitpunkt der Wiederherstellung.

Beispiel: Angenommen, der bereitgestellte Durchsatz einer Tabelle wurde jüngst auf 50 Lese- und 50 Schreibkapazitätseinheiten verringert. Anschließend stellen Sie den Status der Tabelle vor drei Wochen wieder her. Zu diesem Zeitpunkt wurde der bereitgestellte Durchsatz auf 100 Lesekapazitätseinheiten und 100 Schreibkapazitätseinheiten festgelegt. In diesem Fall stellt Amazon Keyspaces Ihre Tabellendaten zu diesem Zeitpunkt wieder her, verwendet jedoch die aktuellen bereitgestellten Durchsatzeinstellungen (50 Lesekapazitätseinheiten und 50 Schreibkapazitätseinheiten).

Die folgenden Einstellungen werden nicht wiederhergestellt, und Sie müssen sie manuell für die neue Tabelle konfigurieren.

- AWS Identity and Access Management (IAM)-Richtlinien
- Amazon- CloudWatch Metriken und Alarmer
- Tags ( kann der CQL-RESTOREAnweisung mit hinzugefügt werden `WITH TAGS`)

## PITR-Wiederherstellung verschlüsselter Tabellen

Wenn Sie eine Tabelle mit PITR wiederherstellen, stellt Amazon Keyspaces die Verschlüsselungseinstellungen Ihrer Quelltable wieder her. Wenn die Tabelle mit einem AWS-eigener Schlüssel (Standard) verschlüsselt wurde, wird die Tabelle automatisch mit derselben Einstellung wiederhergestellt. Wenn die Tabelle, die Sie wiederherstellen möchten, mit einem vom Kunden verwalteten Schlüssel verschlüsselt wurde, muss derselbe vom Kunden verwaltete Schlüssel für Amazon Keyspaces zugänglich sein, um die Tabellendaten wiederherzustellen.

Sie können die Verschlüsselungseinstellungen der Tabelle zum Zeitpunkt der Wiederherstellung ändern. Um von einem zu einem kundenverwalteten Schlüssel AWS-eigener Schlüssel zu wechseln, müssen Sie zum Zeitpunkt der Wiederherstellung einen gültigen und zugänglichen kundenverwalteten Schlüssel angeben.

Wenn Sie von einem vom Kunden verwalteten Schlüssel zu einem wechseln möchtenAWS-eigener Schlüssel, vergewissern Sie sich, dass Amazon Keyspaces Zugriff auf den vom Kunden verwalteten Schlüssel der Quelltable hat, um die Tabelle mit einem wiederherzustellenAWS-eigener Schlüssel. Weitere Informationen zu den Einstellungen für die Verschlüsselung im Ruhezustand für Tabellen finden Sie unter [the section called “Funktionsweise”](#).

#### Note

Wenn die Tabelle gelöscht wurde, weil Amazon Keyspaces den Zugriff auf Ihren vom Kunden verwalteten Schlüssel verloren hat, müssen Sie sicherstellen, dass der vom Kunden verwaltete Schlüssel für Amazon Keyspaces zugänglich ist, bevor Sie versuchen, die Tabelle wiederherzustellen. Eine Tabelle, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt wurde, kann nicht wiederhergestellt werden, wenn Amazon Keyspaces keinen Zugriff auf diesen Schlüssel hat. Weitere Informationen finden Sie unter [Fehlerbehebung beim Schlüsselzugriff im -AWS Key Management ServiceEntwicklerhandbuch](#).

## PITR-Wiederherstellung von multiregionalen Tabellen

Sie können eine multiregionale Tabelle mit PITR wiederherstellen. Damit der Wiederherstellungsvorgang erfolgreich ist, müssen sowohl die Quell- als auch die Zieltabelle in denselben repliziert werdenAWS-Regionen.

Amazon Keyspaces stellt die Einstellungen der Quelltable in jeder der replizierten Regionen wieder her, die Teil des Keyspace sind. Sie können die Einstellungen auch während des Wiederherstellungsvorgangs überschreiben. Weitere Informationen zu Einstellungen, die während der Wiederherstellung geändert werden können, finden Sie unter [the section called “Einstellungen wiederherstellen”](#).

Weitere Informationen zur Multi-Region-Replikation finden Sie unter [the section called “Funktionsweise”](#).

## Tabellenwiederherstellungszeit mit PITR

Die Zeit, die Sie für die Wiederherstellung einer Tabelle benötigen, basiert auf mehreren Faktoren und ist nicht immer direkt mit der Größe der Tabelle korreliert.

Im Folgenden finden Sie einige Überlegungen zu Wiederherstellungszeiten.



- Die Sicherung wird in einer neuen Tabelle wiederhergestellt. Die Ausführung aller mit der Erstellung einer neuen Tabelle verbundenen Aktionen sowie die Initiierung des Wiederherstellungsprozesses kann bis zu 20 Minuten dauern, theoretisch selbst bei einer leeren Tabelle.
- Die Wiederherstellungszeiten für große Tabellen mit gut verteilten Datenmodellen können mehrere Stunden oder länger betragen.
- Wenn Ihre Quelltable Daten enthält, die erheblich verzerrt sind, kann sich die Wiederherstellungszeit verlängern. Wenn der Primärschlüssel Ihrer Tabelle beispielsweise den Monat des Jahres als Partitionsschlüssel verwendet und alle Ihre Daten aus dem Monat Dezember stammen, haben Sie verzerrte Daten.

Eine bewährte Methode bei der Planung der Notfallwiederherstellung besteht darin, die durchschnittlichen Wiederherstellungszeiten regelmäßig zu dokumentieren und festzulegen, wie sich diese Zeiten auf Ihr gesamtes Recovery-Zeitziel auswirken.

## Amazon Keyspaces PITR und Integration in -AWS Services

Die folgenden PITR-Operationen werden mit protokolliert AWS CloudTrail, um eine kontinuierliche Überwachung und Prüfung zu ermöglichen.

- Erstellen Sie eine neue Tabelle mit aktiviertem oder deaktiviertem PITR.
- Aktivieren oder deaktivieren Sie PITR für eine vorhandene Tabelle.
- Stellen Sie eine aktive oder eine gelöschte Tabelle wieder her.

Weitere Informationen finden Sie unter [Protokollieren von Amazon Keyspaces-API-Aufrufen mit AWS CloudTrail](#).

Sie können die folgenden PITR-Aktionen mit ausführen AWS CloudFormation.

- Erstellen Sie eine neue Tabelle mit aktiviertem oder deaktiviertem PITR.
- Aktivieren oder deaktivieren Sie PITR für eine vorhandene Tabelle.

Weitere Informationen finden Sie in der [Cassandra-Ressourcentypreferenz](#) im [AWS CloudFormation - Benutzerhandbuch](#).

# Wiederherstellen einer Amazon Keyspaces-Tabelle auf einen bestimmten Zeitpunkt

Mit Amazon Keyspaces (für Apache Cassandra) point-in-time Recovery (PITR) können Sie Amazon Keyspaces-Tabellendaten zu einem beliebigen Zeitpunkt innerhalb der letzten 35 Tage wiederherstellen. Der erste Teil dieses Tutorials zeigt Ihnen, wie Sie mithilfe der Amazon Keyspaces-Konsole, der () und der Cassandra Query LanguageAWS Command Line Interface (CQLAWS CLI) eine Tabelle auf einen bestimmten Zeitpunkt zurücksetzen. Der zweite Teil zeigt Ihnen, wie Sie eine gelöschte Tabelle mithilfe vonAWS CLI und CQL wiederherstellen.

## Themen

- [Bevor Sie beginnen](#)
- [Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt \(Konsole\)](#)
- [Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit demAWS CLI](#)
- [Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit CQL](#)
- [Wiederherstellen einer gelöschten Tabelle mit demAWS CLI](#)
- [Wiederherstellung einer gelöschten Tabelle mit CQL](#)

## Bevor Sie beginnen

Wenn Sie es noch nicht getan haben, müssen Sie die entsprechenden Berechtigungen für den Benutzer konfigurieren, um Amazon Keyspaces-Tabellen wiederherzustellen. InAWS Identity and Access Management (IAM)AmazonKeyspacesFullAccess beinhaltet dieAWS verwaltete Richtlinie die Berechtigungen zur Wiederherstellung von Amazon Keyspaces-Tabellen. Eine ausführliche Anleitung zur Implementierung einer Richtlinie mit den erforderlichen Mindestberechtigungen finden Sie unter[the section called “Berechtigungen wiederherstellen”](#).


## Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt (Konsole)

Das folgende Beispiel zeigt, wie Sie mit der Amazon Keyspaces-Konsole eine vorhandene Tabelle wiederherstellen,mytable die auf einen bestimmten Zeitpunkt benannt ist.

 Note

Hierbei wird vorausgesetzt, dass Sie die point-in-time Wiederherstellung aktiviert haben. Gehen Sie wie unter beschrieben vor, um PITR für die `mytable` Tabelle zu aktivieren [the section called “Verwenden der Konsole”](#).

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Tables (Tabellen).
3. Wählen Sie in der Tabellenliste die Tabelle `mytable` aus.
4. Wählen Sie auf der Registerkarte Backups der `mytable` Tabelle im Abschnitt point-in-time P-Wiederherstellung die Option Wiederherstellen aus.
5. Geben Sie als neuen Tabellennamen **`mytable_restored`** ein.
6. Um den Zeitpunkt für den Wiederherstellungsvorgang zu definieren, können Sie zwischen zwei Optionen wählen:
  - Wählen Sie die vorkonfigurierte Frühestzeit aus.
  - Wählen Sie „Datum und Uhrzeit angeben“ aus und geben Sie das Datum und die Uhrzeit ein, auf die Sie die neue Tabelle wiederherstellen möchten.

 Note

Sie können die Wiederherstellung auf jeden beliebigen Zeitpunkt innerhalb des frühesten Zeitpunkts und der aktuellen Uhrzeit vornehmen. Amazon Keyspaces stellt Ihre Tabellendaten basierend auf dem ausgewählten Datum und der ausgewählten Uhrzeit (day:minute:second) wieder her.

7. Wählen Sie Wiederherstellen, um den Wiederherstellungsvorgang zu starten.

Die Tabelle, die wiederhergestellt wird, erhält den Status Restoring. Nach Abschluss des Wiederherstellungsvorgangs ändert sich der Status der Tabelle `mytable_restored` in Active (Aktiv).

**⚠ Important**

Modifizieren oder löschen Sie während der Wiederherstellung nicht die AWS Identity and Access Management (IAM)-Richtlinien, die der IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) die Berechtigung zum Ausführen der Wiederherstellung verleihen. Dies könnte andernfalls zu unerwartetem Verhalten führen. Nehmen wir zum Beispiel an, Sie haben die Schreibberechtigungen für eine Tabelle entfernt, während diese Tabelle wiederhergestellt wurde. In diesem Fall kann die zugrunde liegende `RestoreTableToPointInTime`-Operation keine wiederhergestellten Daten in die Tabelle schreiben.

Sie können Berechtigungen erst modifizieren oder löschen, wenn die Wiederherstellung abgeschlossen ist.

## Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit dem AWS CLI

Das folgende Beispiel zeigt, wie Sie mithilfe der AWS CLI die vorhandene Tabelle `myTable` im Zustand eines bestimmten Zeitpunkts wiederherstellen.

1. Im ersten Schritt erstellen Sie eine einfache Tabelle mit dem Namen `myTable`, für die PITR aktiviert ist. Der Befehl wurde aus Gründen der Lesbarkeit in separate Zeilen aufgeteilt.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable'
 --schema-definition 'allColumns=[{name=id,type=int},
{name=name,type=text},{name=date,type=timestamp}],partitionKeys=[{name=id}]'
 --point-in-time-recovery 'status=ENABLED'
```

2. Bestätigen Sie die Eigenschaften der neuen Tabelle und überprüfen Sie die `earliestRestorableTimestamp` für PITR.

```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

Die Ausgabe für den Befehl gibt den folgenden Befehl zurück.

```
{
 "keyspaceName": "myKeyspace",
```

```
"tableName": "myTable",
"resourceArn": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/myKeyspace/
table/myTable",
"creationTimestamp": "2022-06-20T14:34:57.049000-07:00",
"status": "ACTIVE",
"schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
},
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2022-06-20T14:34:57.049000-07:00"
},
"encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
 "status": "ENABLED",
 "earliestRestorableTimestamp": "2022-06-20T14:35:13.693000-07:00"
},
"defaultTimeToLive": 0,
"comment": {
 "message": ""
}
}
```

```
}
```

Sie können eine aktive Tabelle in Intervallen von einer Sekunde auf einen beliebigen Wert `point-in-time` zwischen `derearliestRestorableTimestamp` und der aktuellen Uhrzeit wiederherstellen. Die aktuelle Uhrzeit ist die Standardeinstellung.

3. Geben Sie `a` im Format ISO 8601 an, um eine `Tabellerestore_timestamp` im Zustand eines bestimmten Zeitpunkts wiederherzustellen. Sie können einen beliebigen Zeitpunkt innerhalb der vergangenen 35 Tage im Abstand von einer Sekunde wählen. Beispielsweise wird mit dem folgendem Befehl die Tabelle auf den Zeitpunkt `EarliestRestorableDateTime` wiederhergestellt.

```
aws keyspaces restore-table --source-keyspace-name 'myKeyspace' --source-table-name 'myTable' --target-keyspace-name 'myKeyspace' --target-table-name 'myTable_restored' --restore-timestamp "2022-06-20 21:35:14.693"
```

Die Ausgabe dieses Befehls gibt den ARN der wiederhergestellten Tabelle zurück.

```
{
 "restoredTableARN": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/myKeyspace/table/myTable_restored"
}
```

Um die Tabelle auf die aktuelle Uhrzeit zurückzusetzen, können Sie das `weglassenrestore-timestamp`.

```
aws keyspaces restore-table --source-keyspace-name 'myKeyspace' --source-table-name 'myTable' --target-keyspace-name 'myKeyspace' --target-table-name 'myTable_restored1'"
```

### Important

Modifizieren oder löschen Sie während der Wiederherstellung nicht die AWS Identity and Access Management (IAM)-Richtlinien, die der IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) die Berechtigung zum Ausführen der Wiederherstellung verleihen. Dies könnte andernfalls zu unerwartetem Verhalten führen. Nehmen wir zum Beispiel an, Sie haben die Schreibberechtigungen für eine Tabelle entfernt, während diese Tabelle wiederhergestellt

wurde. In diesem Fall kann die zugrunde liegende `RestoreTableToPointInTime`-Operation keine wiederhergestellten Daten in die Tabelle schreiben. Sie können Berechtigungen erst modifizieren oder löschen, wenn die Wiederherstellung abgeschlossen ist.

## Wiederherstellen einer Tabelle auf einen bestimmten Zeitpunkt mit CQL

Die folgende Prozedur zeigt, wie Sie mit CQL eine vorhandene Tabelle wiederherstellen, `mytable` die auf einen bestimmten Zeitpunkt benannt ist.

### Note

Hierbei wird vorausgesetzt, dass Sie die point-in-time Wiederherstellung aktiviert haben. Um PITR in der Tabelle zu aktivieren, folgen Sie den Schritten unter [the section called "CQL"](#).

1. Sie können eine aktive Tabelle auf einen Wert point-in-time zwischen `earliest_restorable_timestamp` und der aktuellen Uhrzeit wiederherstellen. Die aktuelle Uhrzeit ist die Standardeinstellung.

Um zu überprüfen, ob die point-in-time Wiederherstellung für die `mytable` Tabelle aktiviert ist, fragen Sie `system_schema_mcs.tables` wie folgt ab.

```
SELECT custom_properties
FROM system_schema_mcs.tables
WHERE keyspace_name = 'mykeyspace' AND table_name = 'mytable';
```

Die point-in-time P-Wiederherstellung ist aktiviert, wie die folgende Beispielausgabe zeigt.

```
custom_properties

{
 ...,
 "point_in_time_recovery": {
 "earliest_restorable_timestamp": "2020-06-30T19:19:21.175Z"
 "status": "enabled"
 }
}
```

```
}
```

2. Stellen Sie die Tabelle zu einem Zeitpunkt wieder her, der durch `restore_timestamp` im ISO 8601-Format spezifiziert ist. In diesem Fall wird die `mytable` Tabelle im Zustand des aktuellen Zeitpunkts wiederhergestellt. Sie können die `WITH restore_timestamp = ...` Klausel weglassen. Ohne die Klausel wird der aktuelle Zeitstempel verwendet.

```
RESTORE TABLE mykeyspace.mytable_restored
FROM TABLE mykeyspace.mytable;
```

Sie können auch auf einen bestimmten Zeitpunkt wiederherstellen. Sie können einen beliebigen Zeitpunkt innerhalb der vergangenen 35 Tage angeben. Beispielsweise wird mit dem folgenden Befehl die Tabelle auf den Zeitpunkt `EarliestRestorableDateTime` wiederhergestellt.

```
RESTORE TABLE mykeyspace.mytable_restored
FROM TABLE mykeyspace.mytable
WITH restore_timestamp = '2020-06-30T19:19:21.175Z';
```

Eine vollständige Syntaxbeschreibung finden Sie [the section called “TABELLE WIEDERHERSTELLEN”](#) in der Sprachreferenz.

Um zu überprüfen, ob die Wiederherstellung der Tabelle erfolgreich war, fragen Sie die `ab,system_schema_mcs.tables` um den Status der Tabelle zu bestätigen.

```
SELECT status
FROM system_schema_mcs.tables
WHERE keyspace_name = 'mykeyspace' AND table_name = 'mytable_restored'
```

Die Abfrage zeigt die folgende Ausgabe.

```
status

RESTORING
```

Die Tabelle, die wiederhergestellt wird, erhält den Status `Restoring`. Nach Abschluss des Wiederherstellungsvorgangs ändert sich der Status der Tabelle `mytable_restored` in `Active` (Aktiv).



**⚠ Important**

Modifizieren oder löschen Sie während der Wiederherstellung nicht die AWS Identity and Access Management (IAM)-Richtlinien, die der IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) die Berechtigung zum Ausführen der Wiederherstellung verleihen. Dies könnte andernfalls zu unerwartetem Verhalten führen. Nehmen wir zum Beispiel an, Sie haben die Schreibberechtigungen für eine Tabelle entfernt, während diese Tabelle wiederhergestellt wurde. In diesem Fall kann die zugrunde liegende `RestoreTableToPointInTime`-Operation keine wiederhergestellten Daten in die Tabelle schreiben. Sie können Berechtigungen erst modifizieren oder löschen, wenn die Wiederherstellung abgeschlossen ist.

## Wiederherstellen einer gelöschten Tabelle mit dem AWS CLI

Das folgende Verfahren zeigt, wie Sie die verwenden, AWS CLI um eine gelöschte Tabelle wiederherzustellen, deren Namen `myTable` zum Zeitpunkt des Löschens angegeben sind.

**📘 Note**

Bei diesem Verfahren wird davon ausgegangen, dass PITR für die gelöschte Tabelle aktiviert wurde.

1. Löschen Sie die Tabelle, die Sie im vorherigen Tutorial erstellt haben.

```
aws keyspaces delete-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

2. Stellen Sie mit dem folgenden Befehl den Zeitpunkt des Löschens für die gelöschte Tabelle wieder her.

```
aws keyspaces restore-table --source-keyspace-name 'myKeyspace' --source-table-name 'myTable' --target-keyspace-name 'myKeyspace' --target-table-name 'myTable_restored2'
```

Die Ausgabe dieses Befehls gibt den ARN der wiederhergestellten Tabelle zurück.

```
{
```

```
"restoredTableARN": "arn:aws:cassandra:us-east-1:111222333444:/keyspace/
myKeyspace/table/myTable_restored2"
}
```

## Wiederherstellung einer gelöschten Tabelle mit CQL

Das folgende Verfahren zeigt, wie Sie CQL verwenden, um eine gelöschte Tabelle wiederherzustellen, die `mytable` nach dem Zeitpunkt des Löschens benannt ist.

### Note

Bei diesem Verfahren wird davon ausgegangen, dass PITR für die gelöschte Tabelle aktiviert wurde.

1. Um zu bestätigen, dass die point-in-time Wiederherstellung für eine gelöschte Tabelle aktiviert ist, fragen Sie die Systemtabelle ab. Es werden nur Tabellen mit aktivierter point-in-time Wiederherstellung angezeigt.

```
SELECT custom_properties
FROM system_schema_mcs.tables_history
WHERE keyspace_name = 'mykeyspace' AND table_name = 'my_table';
```

Die Abfrage zeigt die folgende Ausgabe.

```
custom_properties

{
 ...,
 "point_in_time_recovery":{
 "restorable_until_time":"2020-08-04T00:48:58.381Z",
 "status":"enabled"
 }
}
```

2. Stellen Sie mit der folgenden Beispielanweisung den Zeitpunkt des Löschens der Tabelle wieder her.

```
RESTORE TABLE mykeyspace.mytable_restored
```

```
FROM TABLE mykeyspace.mytable;
```

# Ablaufende ende Daten mithilfe Amazon Keyspaces Time to Live

Amazon Keyspaces (für Apache Cassandra) Daten, die Sie nicht mehr benötigen, werden basierend auf dem von Ihnen festgelegten Time-to-Live-Wert automatisch aus Ihrer Tabelle gelöscht. Dies erleichtert die Einhaltung von Richtlinien zur Datenspeicherung, die auf Geschäfts-, Branchen- oder behördlichen Anforderungen basieren und festlegen, wie lange Daten aufbewahrt werden müssen oder wann Daten gelöscht werden müssen.

Sie können TTL beispielsweise in einer AdTech Anwendung verwenden, um zu planen, wann Daten für bestimmte Anzeigen ablaufen und für Kunden nicht mehr sichtbar sind. Sie können TTL auch verwenden, um ältere Daten automatisch zu löschen und Ihre Speicherkosten zu senken. Sie können einen TTL-Standardwert für die gesamte Tabelle festlegen und diesen Wert für einzelne Zeilen und Spalten überschreiben. TTL-Operationen wirken sich nicht auf die Leistung Ihrer Anwendung aus. Außerdem hat die Anzahl der Zeilen und Spalten, die mit TTL als ablaufend markiert sind, keinen Einfluss auf die Verfügbarkeit Ihrer Tabelle.

Amazon Keyspaces filtert abgelaufene Daten automatisch heraus, sodass abgelaufene Daten nicht in Abfrageergebnissen zurückgegeben werden oder für die Verwendung in DML-Anweisungen (Data Manipulation Language) verfügbar sind. Amazon Keyspaces löscht abgelaufene Daten in der Regel innerhalb von 10 Tagen nach dem Ablaufdatum aus dem Speicher. In seltenen Fällen ist Amazon Keyspaces möglicherweise nicht in der Lage, Daten innerhalb von 10 Tagen zu löschen, wenn auf der zugrunde liegenden Speicherpartition anhaltende Aktivitäten stattfinden, um die Verfügbarkeit zu schützen. In diesen Fällen versucht Amazon Keyspaces weiterhin, die abgelaufenen Daten zu löschen, sobald der Traffic auf der Partition abnimmt. Nachdem die Daten dauerhaft aus dem Speicher gelöscht wurden, fallen keine Speichergebühren mehr an. Weitere Informationen finden Sie unter [the section called “Funktionsweise”](#).

Sie können TTL-Standardinstellungen für neue und bestehende Tabellen festlegen, ändern oder deaktivieren, indem Sie die Konsole oder Cassandra Query Language (CQL) verwenden. In Tabellen mit konfigurierter TTL-Standardinstellung können Sie Cassandra Query Language (CQL) verwenden, um die TTL-Standardinstellungen zu überschreiben und benutzerdefinierte TTL-Werte auf Zeilen und Spalten anzuwenden. Weitere Informationen finden Sie unter [the section called “Verwenden von Time to Live to Live”](#).

Die TTL-Preise basieren auf der Größe der Zeilen, die mithilfe von Time to Live gelöscht oder aktualisiert werden. TTL-Operationen werden in Einheiten von gemessenTTL deletes. Ein

TTL-Löschvorgang wird pro KB Daten pro gelöschter oder aktualisierter Zeile verbraucht. Um beispielsweise eine Zeile zu aktualisieren, die 2,5 KB Daten speichert, und um eine oder mehrere Spalten innerhalb der Zeile gleichzeitig zu löschen, sind drei TTL-Löschungen erforderlich. Oder um eine ganze Zeile zu löschen, die 3,5 KB Daten enthält, sind vier TTL-Löschungen erforderlich. Ein TTL-Löschvorgang wird pro KB gelöschter Daten pro Zeile verbraucht. Weitere Informationen zu Preisen finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\)](#)

#### Themen

- [Funktionsweise des Amazon Keyspaces Time to Live \(TTL\) von Amazon Keyspaces Time to Live \(TTL\) von Amazon Keyspace Keyspaces](#)
- [Verwendung von für Gültigkeitsdauer](#)

## Funktionsweise des Amazon Keyspaces Time to Live (TTL) von Amazon Keyspaces Time to Live (TTL) von Amazon Keyspace Keyspaces

Amazon Keyspaces Time to Live (TTL) von Amazon Keyspaces wird vollständig verwaltet. Sie müssen sich nicht um Systemeinstellungen auf niedriger Ebene wie Komprimierungsstrategien kümmern. Daten laufen zu dem von Ihnen angegebenen Zeitpunkt ab, und Amazon Keyspaces entfernt abgelaufene Daten automatisch (in der Regel innerhalb von 10 Tagen), ohne die Leistung oder Verfügbarkeit Ihrer Anwendung zu beeinträchtigen.

Abgelaufene Daten sind zum Löschen markiert und stehen für DML-Anweisungen (Data Manipulation Language nicht zur Datenmanipulation Language zur Verfügung. Wenn Sie weiterhin Lese- und Schreibvorgänge für Zeilen ausführen, die abgelaufene Daten enthalten, werden die abgelaufenen Daten weiterhin auf die Lesekapazitätseinheiten (RCUs) und Schreibkapazitätseinheiten (WCUs) angerechnet, bis sie aus dem Speicher gelöscht werden.

#### Themen

- [Festlegen des TTL-Standardwerts für eine Tabelle](#)
- [Festlegen benutzerdefinierter TTL-Werte für Zeilen und Spalten](#)
- [TTL für Tabellen aktivieren](#)
- [Amazon Keyspaces Time to Live und Integration mitAWS Services](#)

## Festlegen des TTL-Standardwerts für eine Tabelle

In Amazon Keyspaces können Sie bei der Erstellung der Tabelle einen Standard-TTL-Wert für alle Zeilen in einer Tabelle festlegen. Sie können auch eine vorhandene Tabelle bearbeiten, um den Standard-TTL-Wert für neue in die Tabelle eingefügte Zeilen festzulegen oder zu ändern. Durch das Ändern des Standard-TTL-Werts einer Tabelle wird der TTL-Wert vorhandener Daten in der Tabelle nicht geändert. Der Standard-TTL-Wert für eine Tabelle ist Null ist, was bedeutet, dass Daten nicht automatisch ablaufen als Nächstes ablaufen lassen. Wenn der TTL-Standardwert für eine Tabelle größer als Null ist, wird jeder Zeile ein Ablaufzeitstempel hinzugefügt.

Amazon Keyspaces berechnet bei jeder Aktualisierung der Daten einen neuen TTL-Zeitstempel. TTL-Werte werden in Sekunden festgelegt, und der maximal konfigurierbare Wert ist 630 720 000 Sekunden beträgt, was 20 Jahren entspricht. Weitere Hinweise zum Festlegen, Ändern und Deaktivieren des TTL-Standardwerts für Tabellen, die den AWS Management Console oder CQL verwenden, finden Sie unter [the section called “Verwenden von Time to Live to Live”](#).

## Festlegen benutzerdefinierter TTL-Werte für Zeilen und Spalten

### Note

Bevor Sie benutzerdefinierte TTL-Werte für Zeilen und Spalten festlegen, muss TTL zuerst für die Tabelle aktiviert werden. Weitere Informationen finden Sie unter [the section called “Aktivieren von für Gültigkeitsdauer \(TTL\)”](#).

Um den TTL-Standardwert einer Tabelle zu überschreiben oder Verfallsdaten für einzelne Zeilen festzulegen, können Sie die folgenden DML-Anweisungen (Data Manipulation Language) in CQL verwenden:

- INSERT— Dient zum Einfügen einer neuen Datenzeile mit einem TTL-Wertesatz.
- UPDATE— Wird verwendet, um eine bestehende Datenzeile mit einem neuen TTL-Wert zu ändern.

Das Festlegen von TTL-Werten für Zeilen hat Vorrang vor der TTL-StandardEinstellung für die Tabelle.

Für die Caching-Methode und Beispiele vgl. [the section called “Verwenden INSERT von für Gültigkeitsdauer to Live\)”](#).

Um TTL-Werte für einzelne Spalten zu überschreiben oder festzulegen, können Sie die TTL-Einstellung für eine Teilmenge von Spalten innerhalb vorhandener Zeilen aktualisieren, indem Sie die folgende CQL-DML-Anweisung verwenden:

- `UPDATE`— Wird verwendet, um eine Datenspalte zu aktualisieren.

Das Festlegen von TTL-Werten für Spalten hat Vorrang vor der TTL-StandardEinstellung für die Tabelle und allen benutzerdefinierten TTL-Einstellungen für die Zeile. Für die Caching-Methode und Beispiele vgl. [vgl. the section called “Verwenden UPDATE von für Gültigkeitsdauer to Live\)”](#).

## TTL für Tabellen aktivieren

TTL wird automatisch für Tabellen aktiviert, wenn Sie in einer der `ALTER TABLE` Anweisungen `CREATE TABLE` oder ein `default_time_to_live` Wert größer als 0 angeben. Wenn Sie `default_time_to_live` für die Tabelle keinen Wert angeben, aber mithilfe `INSERT` von oder `UPDATE` -Operationen benutzerdefinierte TTL-Werte für Zeilen oder Spalten angeben möchten, müssen Sie zuerst TTL für die Tabelle aktivieren. Mithilfe der `t1` benutzerdefinierten Eigenschaft können Sie TTL für eine Tabelle aktivieren.

Wenn Sie TTL für eine Tabelle aktivieren, beginnt Amazon Keyspaces, zusätzliche TTL-bezogene Metadaten für jede Zeile zu speichern. Darüber hinaus verwendet TTL Ablaufzeitstempel, um zu verfolgen, wann Zeilen oder Spalten ablaufen. Die Zeitstempel werden als Zeilenmetadaten gespeichert und tragen zu den Speicherkosten für die Zeile bei.

Nachdem Sie die TTL-Funktion aktiviert haben, können Sie sie für eine Tabelle nicht mehr deaktivieren. Wenn Sie die `default_time_to_live` auf 0 setzen, werden die standardmäßigen Ablaufzeiten für neue Daten deaktiviert, die TTL-Funktion wird dadurch jedoch nicht deaktiviert und die Tabelle wird nicht auf die ursprünglichen Speichermetadaten oder das Schreibverhalten von Amazon Keyspaces zurückgesetzt.

## Amazon Keyspaces Time to Live und Integration mit AWS Services

Die folgende TTL-Metrik ist in Amazon verfügbar CloudWatch , um eine kontinuierliche Überwachung zu ermöglichen.

- `TTLDeletes`— Die Einheiten, die zum Löschen oder Aktualisieren von Daten hintereinander mithilfe von Time to Live (TTL) verbraucht wurden.

Weitere Informationen zur Überwachung der CloudWatch -Metriken finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

Wenn Sie es verwenden AWS CloudFormation, können Sie TTL aktivieren, wenn Sie eine Amazon Keyspaces-Tabelle erstellen. Weitere Informationen finden Sie im [AWS CloudFormation-Benutzerhandbuch](#).

## Verwendung von für Gültigkeitsdauer

Sie können die Amazon-Keyspaces (für Apache Cassandra) Konsole oder CQL verwenden, um die Einstellungen für Apache Cassandra).

### Themen

- [Erstellen einer neuen Tabelle mit aktivierten TTL-Einstellungen \(Konsole\)](#)
- [Um die TTL-Einstellungen \(für Gültigkeitsdauer\)](#)
- [Deaktivieren von für Gültigkeitsdauer to Live\)](#)
- [Erstellen einer neuen Tabelle mit den TTL-Einstellungen \(für Gültigkeitsdauer\)](#)
- [Verwenden ALTER TABLE von für Gültigkeitsdauer to Live\)](#)
- [Aktivieren von für Gültigkeitsdauer \(TTL\)](#)
- [Aktivieren von für Gültigkeitsdauer \(TTL\)](#)
- [Verwenden INSERT von für Gültigkeitsdauer to Live\)](#)
- [Verwenden UPDATE von für Gültigkeitsdauer to Live\)](#)

## Erstellen einer neuen Tabelle mit aktivierten TTL-Einstellungen (Konsole)

Führen Sie diese Schritte aus, um eine neue Tabelle mit den Amazon Keyspaces Konsole zu erstellen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-Keyspaces Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tables (Tabellen) und anschließend Create table (Tabelle erstellen) aus.
3. Wählen Sie auf der Seite Tabelle erstellen im Abschnitt Tabellendetails einen Schlüsselraum aus und geben Sie einen Namen für die neue Tabelle ein.
4. Erstellen Sie im Abschnitt Schema das Schema für Ihre Tabelle.



5. Wählen Sie im Abschnitt Tabelleneinstellungen die Option Einstellungen anpassen aus.
6. Fahren Sie mit für Gültigkeitsdauer to Live)

In diesem Schritt wählen Sie die TTL-Einstellungen für die Tabelle aus.

Geben Sie für den Standard-TTL-Periode die Ablaufzeit ein und wählen Sie die von Ihnen eingegebene Zeiteinheit aus, zum Beispiel Sekunden, Tage oder Jahre. Amazon Keyspaces speichert den Wert in Sekunden.

7. Wählen Sie Create table (Tabelle erstellen) aus. Ihre Tabelle wird mit dem angegebenen Standard-TTL-Wert erstellt.

#### Note

Sie können die Standard-TTL-Einstellung der Tabelle für bestimmte Zeilen oder Spalten überschreiben, indem Sie die Datenmanipulationssprache (DML) im CQL-Editor verwenden.

## Um die TTL-Einstellungen (für Gültigkeitsdauer)

Gehen Sie wie folgt vor, um die Time to Live-Einstellungen für bestehende Tabellen mithilfe der Amazon Keyspaces-Konsole zu aktualisieren.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-Keyspaces Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie die Tabelle, die Sie aktualisieren möchten. Wählen Sie dann die Registerkarte Zusätzliche Einstellungen.
3. Fahren Sie mit Time to Live (TTL) fort und wählen Sie Bearbeiten.
4. Geben Sie für den Standard-TTL-Periode die Ablaufzeit ein und wählen Sie die von Ihnen eingegebene Zeiteinheit aus, zum Beispiel Sekunden, Tage oder Jahre. Amazon Keyspaces speichert den Wert in Sekunden. Dadurch wird der TTL-Wert vorhandener Zeilen nicht geändert.
5. Wenn die TTL-Einstellungen definiert sind, wählen Sie Änderungen speichern.

## Deaktivieren von für Gültigkeitsdauer to Live)

Gehen Sie wie folgt vor, um die Time to Live-Einstellungen für bestehende Tabellen mithilfe der Amazon Keyspaces AWS Management Console zu deaktivieren.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-Keyspaces Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie die Tabelle, die Sie aktualisieren möchten. Wählen Sie dann die Registerkarte Zusätzliche Einstellungen.
3. Fahren Sie mit Time to Live (TTL) fort und wählen Sie Bearbeiten.
4. Wählen Sie Standard-TTL-Periode und setzen Sie den Wert auf Null. Dadurch wird TTL für die Tabelle standardmäßig für future Daten deaktiviert. Der TTL-Wert für bestehende Zeilen wird dadurch nicht geändert.
5. Wenn die TTL-Einstellungen definiert sind, wählen Sie Änderungen speichern.

## Erstellen einer neuen Tabelle mit den TTL-Einstellungen (für Gültigkeitsdauer)

Aktivieren Sie TTL, wenn Sie eine neue Tabelle erstellen, deren Standard-TTL-Wert auf 3.024.000 Sekunden festgelegt ist, was 35 Tagen entspricht.

```
CREATE TABLE my_table (
 userid uuid,
 time timeuuid,
 subject text,
 body text,
 user inet,
 PRIMARY KEY (userid, time)
) WITH default_time_to_live = 3024000;
```

Verwenden Sie die `cqlsh describe` Anweisung, wie im folgenden Beispiel gezeigt, um die TTL-Einstellungen für die neue Tabelle zu bestätigen. Die Ausgabe zeigt die Standard-TTL-Einstellung für die Tabelle als `default_time_to_live`.

```
describe my_table;
```

## Verwenden **ALTER TABLE** von für Gültigkeitsdauer to Live)

Aktualisieren Sie die TTL-Einstellungen der vorhandenen Tabelle auf 2.592.000 Sekunden, was 30 Tagen entspricht.

```
ALTER TABLE my_table WITH default_time_to_live = 2592000;
```

Verwenden Sie die `cqlsh describe` Anweisung, wie im folgenden Beispiel gezeigt, um die TTL-Einstellungen für die aktualisierte Tabelle zu bestätigen. Die Ausgabe zeigt die Standard-TTL-Einstellung für die Tabelle `alsdefault_time_to_live`.

```
describe my_table;
```

## Aktivieren von für Gültigkeitsdauer (TTL)

Um benutzerdefinierte Time to Live-Einstellungen zu aktivieren, die auf Zeilen und Spalten angewendet werden können, ohne die TTL-Standard-Einstellungen für die gesamte Tabelle zu aktivieren, können Sie die folgende CQL-Anweisung verwenden.

```
CREATE TABLE my_keyspace.my_table (id int primary key) WITH CUSTOM_PROPERTIES={'ttl': {'status': 'enabled'}};
```

Wenn `ttl` aktiviert, können Sie sie für die Tabelle nicht mehr deaktivieren.

## Aktivieren von für Gültigkeitsdauer (TTL)

Um benutzerdefinierte Time to Live-Einstellungen zu aktivieren, die auf Zeilen und Spalten angewendet werden können, ohne die TTL-Standard-Einstellungen für die gesamte Tabelle zu aktivieren, können Sie die folgende CQL-Anweisung verwenden.

```
ALTER TABLE my_table WITH CUSTOM_PROPERTIES={'ttl':{'status': 'enabled'}};
```

Wenn `ttl` aktiviert, können Sie sie für die Tabelle nicht mehr deaktivieren.

## Verwenden **INSERT** von für Gültigkeitsdauer to Live)

Die folgende CQL-Anweisung fügt eine Datenzeile in die Tabelle ein und ändert die Standard-TTL-Einstellung auf 259.200 Sekunden (was 3 Tagen entspricht).

```
INSERT INTO my_table (userid, time, subject, body, user)
 VALUES (B79CB3BA-745E-5D9A-8903-4A02327A7E09, 96a29100-5e25-11ec-90d7-
b5d91eceda0a, 'Message', 'Hello', '205.212.123.123')
 USING TTL 259200;
```

Verwenden Sie die folgende Anweisung, um die TTL-Einstellungen für die eingefügte Zeile zu bestätigen.

```
SELECT TTL (subject) from my_table;
```

## Verwenden **UPDATE** von für Gültigkeitsdauer to Live)

Verwenden Sie die folgende Anweisung, um die TTL-Einstellungen der zuvor eingefügten Spalte „Betreff“ von 259.200 Sekunden (3 Tage) auf 86.400 Sekunden (ein Tag) zu ändern.

```
UPDATE my_table USING TTL 86400 set subject = 'Updated Message' WHERE userid =
 B79CB3BA-745E-5D9A-8903-4A02327A7E09 and time = 96a29100-5e25-11ec-90d7-b5d91eceda0a;
```

Sie können eine einfache Auswahlabfrage ausführen, um den aktualisierten Datensatz vor Ablauf der Gültigkeitsdauer zu sehen.

```
SELECT * from my_table;
```

Die Abfrage zeigt die folgende Ausgabe.

| userid                               | subject         | user            | time                                 | body  |
|--------------------------------------|-----------------|-----------------|--------------------------------------|-------|
| b79cb3ba-745e-5d9a-8903-4a02327a7e09 | Updated Message | 205.212.123.123 | 96a29100-5e25-11ec-90d7-b5d91eceda0a | Hello |
| 50554d6e-29bb-11e5-b345-feff819cdc9f | Message         | 205.212.123.123 | cf03fb21-59b5-11ec-b371-dff626ab9620 | Hello |

Um zu bestätigen, dass der Ablauf erfolgreich war, führen Sie dieselbe Abfrage nach der konfigurierten Ablaufzeit erneut aus.

```
SELECT * from my_table;
```

Die Abfrage zeigt die folgende Ausgabe, nachdem die Spalte „Betreff“ abgelaufen ist.

| userid                               | subject | user            | time                                 | body  |
|--------------------------------------|---------|-----------------|--------------------------------------|-------|
| b79cb3ba-745e-5d9a-8903-4a02327a7e09 | null    | 205.212.123.123 | 96a29100-5e25-11ec-90d7-b5d91eceda0a | Hello |

```
50554d6e-29bb-11e5-b345-feff819cdc9f | cf03fb21-59b5-11ec-b371-dff626ab9620 | Hello |
Message | 205.212.123.123
```

# Arbeiten mit clientseitigen Zeitstempeln in Amazon Keyspaces

In Amazon Keyspaces sind clientseitige Zeitstempel Cassandra-kompatible Zeitstempel, die für jede Zelle in Ihrer Tabelle beibehalten werden. Sie können clientseitige Zeitstempel für die Konfliktlösung verwenden, indem Sie Ihre Client-Anwendungen die Reihenfolge der Schreibvorgänge festlegen lassen. Wenn beispielsweise Clients einer global verteilten Anwendung dieselben Daten aktualisieren, behalten die clientseitigen Zeitstempel die Reihenfolge bei, in der die Aktualisierungen auf den Clients vorgenommen wurden. Amazon Keyspaces verwendet diese Zeitstempel, um die Schreibvorgänge zu verarbeiten. Weitere Informationen finden Sie unter [the section called “Funktionsweise”](#).

Nachdem clientseitige Zeitstempel für eine Tabelle aktiviert wurden, können Sie einen Zeitstempel mit der `USING TIMESTAMP` Klausel in Ihrer Data Manipulation Language (DML) -CQL-Abfrage angeben. Wenn Sie in Ihrer CQL-Abfrage keinen Zeitstempel angeben, verwendet Amazon Keyspaces den von Ihrem Client-Treiber übergebenen Zeitstempel. Wenn der Client-Treiber keine Zeitstempel bereitstellt, weist Amazon Keyspaces automatisch einen Zeitstempel auf Zellebene zu. Um nach Zeitstempeln abzufragen, können Sie die `WRITETIME` Funktion in Ihrer DML-Anweisung verwenden. Weitere Informationen finden Sie unter [the section called “Wie man die Zeitstempel in Betracht ziehen kann”](#).

Amazon Keyspaces berechnet keine zusätzlichen Gebühren, um clientseitige Zeitstempel zu aktivieren. Mit clientseitigen Zeitstempeln speichern und schreiben Sie jedoch zusätzliche Daten für jeden Wert in Ihrer Zeile. Dies kann zu zusätzlichem Speicherverbrauch und in einigen Fällen zu zusätzlichem Durchsatz führen. Weitere Informationen zum Abschätzen der Auswirkungen auf die Zeilengröße finden Sie unter [the section called “Funktionsweise”](#). Weitere Informationen zur Preisgestaltung von Amazon Keyspaces finden Sie unter [Amazon Keyspaces \(für Apache Cassandra\)](#).

## Themen

- [So funktionieren clientseitige Zeitstempel in Amazon Keyspaces](#)
- [Verwenden der Website in Amazon Keyspaces](#)

## So funktionieren clientseitige Zeitstempel in Amazon Keyspaces

Die clientseitigen Zeitstempel von Amazon Keyspace werden vollständig verwaltet. Sie müssen sich nicht um grundlegende Systemeinstellungen wie Bereinigungs- und Verdichtungsstrategien kümmern.

Wenn Sie Daten löschen, werden die Zeilen zum Löschen mit einem Grabstein markiert. Amazon Keyspaces entfernt veraltete Daten automatisch (in der Regel innerhalb von 10 Tagen), ohne die Leistung oder Verfügbarkeit Ihrer Anwendung zu beeinträchtigen. Für DML-Anweisungen (DML Data Manipulation Language) stehen DML-Anweisungen (DML) nicht zur DatenManipulation Language (DML) zur Verfügung. Wenn Sie weiterhin Lese- und Schreibvorgänge für Zeilen ausführen, die veraltete Daten enthalten, werden die veralteten Daten weiterhin für Speicher, Lesekapazitätseinheiten (RCUs) und Schreibkapazitätseinheiten (WCUs) angerechnet, bis sie aus dem Speicher gelöscht werden.

### Themen

- [So funktionieren clientseitige Zeitstempel in Amazon Keyspaces](#)
- [Client-seitige Zeitstempel von Amazon Keyspace und Integration mit AWS Services](#)

## So funktionieren clientseitige Zeitstempel in Amazon Keyspaces

Wenn clientseitige Zeitstempel in Amazon Keyspaces aktiviert sind, speichert jede Spalte jeder Zeile einen Zeitstempel. Diese Zeitstempel beanspruchen ungefähr 20–40 Byte (abhängig von Ihren Daten) und tragen zu den Speicher- und Durchsatzkosten für die Zeile bei. Diese Metadaten-Bytes zählen auch zu Ihrem Kontingent für die Zeilengröße von 1 MB. Um die Gesamtzunahme des Speicherplatzes zu ermitteln (um sicherzustellen, dass die Zeilengröße unter 1 MB bleibt), berücksichtigen Sie die Anzahl der Spalten in Ihrer Tabelle und die Anzahl der Sammlungselemente in jeder Zeile. Wenn eine Tabelle beispielsweise 20 Spalten hat, wobei jede Spalte 40 Byte an Daten speichert, erhöht sich die Größe der Zeile von 800 Byte auf 1200 Byte. Weitere Informationen zum Schätzen der Zeilengröße finden Sie unter [the section called “Berechnung der Zeilengröße”](#). Zusätzlich zu den zusätzlichen 400 Byte für Speicher erhöht sich in diesem Beispiel die Anzahl der pro Schreibvorgang verbrauchten Schreibkapazitätseinheiten (WCUs) von 1 WCU auf 2 WCUs. Weitere Hinweise zur Berechnung der Lese- und Schreibkapazität finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#).

Nachdem die clientseitigen Zeitstempel für eine Tabelle aktiviert wurden, können Sie sie nicht mehr ausschalten. Außerdem können Zeitstempel nicht verwendet werden NULL. Wenn also keine

clientseitigen Zeitstempel von der CQL-Anweisung oder dem Client-Treiber bereitgestellt werden, wird automatisch ein von Amazon Keyspaces generierter Zeitstempel hinzugefügt.

## Client-seitige Zeitstempel von Amazon Keyspace und Integration mit AWS Services

Die folgende clientseitige Zeitstempelmetrik ist in Amazon verfügbar CloudWatch , um eine kontinuierliche Überwachung zu ermöglichen.

- `SystemReconciliationDeletes`— Die Anzahl der Löschvorgänge, die erforderlich sind, um veraltete Daten zu entfernen.

Weitere Informationen zur Überwachung CloudWatch -Metriken finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

## Verwenden der Website in Amazon Keyspaces

Sie können die Amazon Keyspaces-Konsole (für Apache Cassandra), die Cassandra Query Language (CQL), das AWS SDK und die AWS Command Line Interface (AWS CLI) verwenden, um clientseitige Zeitstempel zu aktivieren. Dieser Abschnitt enthält Beispiele dafür, wie clientseitige Zeitstempel für neue und bestehende Tabellen aktiviert werden und wie clientseitige Zeitstempel in Abfragen verwendet werden. Weitere Informationen zur -API finden Sie in der [Amazon Keyspaces-API-Referenz](#).

### Important

Client-seitige Zeitstempel können nicht deaktiviert werden. Das Aktivieren von clientseitigen Zeitstempeln ist eine einmalige Änderung. Amazon Keyspaces bietet keine Option, um es zu deaktivieren, ohne die Tabelle zu löschen.

### Themen

- [Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln \(Konsole\)](#)
- [Clientseitige Zeitstempel für bestehende Tabellen aktivieren \(Konsole\)](#)
- [Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln \(CQL\)](#)
- [Clientseitige Zeitstempel für bestehende Tabellen mithilfe von ALTER TABLE \(CQL\) aktivieren](#)



- [Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln \(CLI\)](#)
- [Clientseitige Zeitstempel für eine vorhandene Tabelle aktivieren \(CLI\)](#)
- [Verwendung von clientseitigen Zeitstempeln in DML-Anweisungen \(Data Manipulation Language\)](#)

## Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln (Konsole)

Führen Sie diese Schritte aus, um eine neue Tabelle zu erstellen, in der Amazon Keyspaces-Konsole eine neue Tabelle zu erstellen.

Um eine neue Tabelle mit clientseitigen Zeitstempeln zu erstellen (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tables (Tabellen) und anschließend Create table (Tabelle erstellen) aus.
3. Wählen Sie auf der Seite Tabelle erstellen im Abschnitt Tabellendetails ein Schlüsselfeld aus und geben Sie einen Namen für die neue Tabelle ein.
4. Erstellen Sie im Abschnitt Schema das Schema für Ihre Tabelle.
5. Wählen Sie im Abschnitt Tabelleneinstellungen die Option Einstellungen anpassen aus.
6. Fahren Sie mit den clientseitigen Zeitstempeln fort.

Wählen Sie Clientseitige Zeitstempel aktivieren, um clientseitige Zeitstempel für die Tabelle zu aktivieren.

7. Wählen Sie Create table (Tabelle erstellen) aus. Ihre Tabelle wird mit aktivierten clientseitigen Zeitstempeln erstellt.

## Clientseitige Zeitstempel für bestehende Tabellen aktivieren (Konsole)

Gehen Sie wie folgt vor, um clientseitige Zeitstempel für bestehende Tabellen mithilfe der Amazon Keyspaces zu aktivieren AWS Management Console.

Um clientseitige Zeitstempel für eine bestehende Tabelle zu aktivieren (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.

2. Wählen Sie die Tabelle aus, die Sie aktualisieren möchten. Wählen Sie dann die Registerkarte Weitere Einstellungen.
3. Gehen Sie auf der Registerkarte Zusätzliche Einstellungen zu Clientseitige Zeitstempel ändern und wählen Sie Clientseitige Zeitstempel aktivieren
4. Wählen Sie Änderungen speichern, um die Einstellungen der Tabelle zu ändern.

## Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln (CQL)

Um clientseitige Zeitstempel zu aktivieren, wenn Sie eine neue Tabelle erstellen, können Sie die folgende CQL-Anweisung verwenden.

```
CREATE TABLE my_table (
 userid uuid,
 time timeuuid,
 subject text,
 body text,
 user inet,
 PRIMARY KEY (userid, time)
) WITH CUSTOM_PROPERTIES = {'client_side_timestamps': {'status': 'enabled'}};
```

Um die clientseitigen Zeitstemeleinstellungen für die neue Tabelle zu bestätigen, verwenden Sie eine SELECT Anweisung, um die zu überprüfen, `custom_properties` wie im folgenden Beispiel gezeigt.

```
SELECT custom_properties from system_schema_mcs.tables where keyspace_name =
'my_keyspace' and table_name = 'my_table';
```

Die Ausgabe dieser Anweisung zeigt den Status für clientseitige Zeitstempel.

```
'client_side_timestamps': {'status': 'enabled'}
```

## Clientseitige Zeitstempel für bestehende Tabellen mithilfe von **ALTER TABLE** (CQL) aktivieren

Um clientseitige Zeitstempel für eine vorhandene Tabelle zu aktivieren, können Sie die folgende CQL-Anweisung verwenden.

```
ALTER TABLE my_table WITH custom_properties = {'client_side_timestamps': {'status': 'enabled'}};
```

Um die clientseitigen Zeitstempelinstellungen für die neue Tabelle zu bestätigen, verwenden Sie eine SELECT Anweisung, um die zu überprüfen, `custom_properties` wie im folgenden Beispiel gezeigt.

```
SELECT custom_properties from system_schema_mcs.tables where keyspace_name = 'my_keyspace' and table_name = 'my_table';
```

Die Ausgabe dieser Anweisung zeigt den Status für clientseitige Zeitstempel.

```
'client_side_timestamps': {'status': 'enabled'}
```

## Erstellen einer neuen Tabelle mit aktivierten clientseitigen Zeitstempeln (CLI)

Wenn Sie eine neue Tabelle erstellen, können Sie die folgende CLI-Anweisung zu einem für die Erstellung einer neuen Tabelle auswählen.

```
./aws keyspaces create-table \
--keyspace-name my_keyspace \
--table-name my_table \
--client-side-timestamps 'status=ENABLED' \
--schema-definition 'allColumns=[{name=id,type=int},{name=date,type=timestamp},
{name=name,type=text}],partitionKeys=[{name=id}]'
```

Führen Sie den folgenden Code aus, um zu überprüfen, ob die clientseitigen Zeitstempel für die neue Tabelle aktiviert sind.

```
./aws keyspaces get-table \
--keyspace-name my_keyspace \
--table-name my_table
```

Die Ausgabe sollte in Betracht ziehen, wie in diesem Beispiel.

```
{
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
```

```
"resourceArn": "arn:aws:cassandra:us-east-2:555555555555:/keyspace/my_keyspace/
table/my_table",
"creationTimestamp": 1662681206.032,
"status": "ACTIVE",
"schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
},
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1662681206.032
},
"encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
 "status": "DISABLED"
},
"clientSideTimestamps": {
 "status": "ENABLED"
},
"ttl": {
 "status": "ENABLED"
},
"defaultTimeToLive": 0,
```

```
 "comment": {
 "message": ""
 }
 }
}
```

## Clientseitige Zeitstempel für eine vorhandene Tabelle aktivieren (CLI)

Um clientseitige Zeitstempel für eine vorhandene Tabelle mithilfe der CLI zu aktivieren, können Sie den folgenden Code verwenden.

```
./aws keyspaces update-table \
--keyspace-name my_keyspace \
--table-name my_table \
--client-side-timestamps 'status=ENABLED'
```

Führen Sie den folgenden Code aus, um zu überprüfen, ob die clientseitigen Zeitstempel für die Tabelle aktiviert sind.

```
./aws keyspaces get-table \
--keyspace-name my_keyspace \
--table-name my_table
```

Die Ausgabe sollte in Betracht ziehen, wie in diesem Beispiel.

```
{
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "resourceArn": "arn:aws:cassandra:us-east-2:555555555555:/keyspace/my_keyspace/
table/my_table",
 "creationTimestamp": 1662681312.906,
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 }
]
 }
}
```

```

 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
},
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1662681312.906
},
"encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
 "status": "DISABLED"
},
"clientSideTimestamps": {
 "status": "ENABLED"
},
"ttl": {
 "status": "ENABLED"
},
"defaultTimeToLive": 0,
"comment": {
 "message": ""
}
}

```

## Verwendung von clientseitigen Zeitstempeln in DML-Anweisungen (Data Manipulation Language)

Nachdem Sie die clientseitigen Zeitstempel aktiviert haben, können Sie den Zeitstempel in Ihren `INSERT`, `UPDATE`, und `DELETE` -Anweisungen zusammen mit der `USING TIMESTAMP` Klausel übergeben. Der Zeitstempelwert steht für die Anzahl von Mikrosekunden seit der Standardbasiszeit, bekannt als epoch: 1. Januar 1970 um 00:00:00 GMT. Ein vom Kunden

bereitgestellter Zeitstempel muss zwischen dem Bereich von 2 Tagen in der Vergangenheit und 5 Minuten in der future gegenüber der aktuellen Wanduhr liegen. Amazon Keyspaces bewahrt die Metadaten der Zeitstempel für die Nutzungsdauer der Daten auf. Sie können die `WRITETIME` Funktion verwenden, um nach Zeitstempeln zu suchen, die in der Vergangenheit aufgetreten sind. Weitere Informationen zur CQL-Syntax finden Sie unter [the section called “DML-Anweisungen.”](#)

Die folgende CQL-Anweisung ist ein Beispiel für die Verwendung eines Zeitstempels als `update_parameter`.

```
INSERT INTO catalog.book_awards (year, award, rank, category, book_title, author, publisher)
VALUES (2022, 'Wolf', 4, 'Non-Fiction', 'Science Update', 'Ana Carolina Silva', 'SomePublisher')
USING TIMESTAMP 1669069624;
```

Wenn Sie in Ihrer CQL-Abfrage keinen Zeitstempel angeben, verwendet Amazon Keyspaces den von Ihrem Client-Treiber übergebenen Zeitstempel. Wenn der Client-Treiber keinen Zeitstempel bereitstellt, weist Amazon Keyspaces Ihrem Schreibvorgang einen serverseitigen Zeitstempel zu.

Um den Zeitstempelwert zu sehen, der für eine bestimmte Spalte gespeichert ist, können Sie die `WRITETIME` Funktion in einer `SELECT` Anweisung verwenden, wie im folgenden Beispiel gezeigt.

```
SELECT year, award, rank, category, book_title, author, publisher, WRITETIME(year),
WRITETIME(award), WRITETIME(rank),
WRITETIME(category), WRITETIME(book_title), WRITETIME(author), WRITETIME(publisher)
from catalog.book_awards;
```

# Amazon Keyspaces-Ressourcen erstellen mitAWS CloudFormation

Amazon Keyspaces (for Apache Cassandra) ist in integriert. Dies ist ein ServiceAWS CloudFormation, der Ihnen hilft, IhreAWS -Ressourcen zu modellieren und einzurichten, so dass Sie weniger Zeit für die Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur aufwenden müssen. Sie erstellen eine Vorlage, in der alle gewünschtenAWS -Ressourcen (wie etwa Schlüsselräume und Tabellen), undAWS CloudFormation übernimmt die Bereitstellung und Konfigurierung dieser Ressourcen für Sie.

Wenn Sie verwendenAWS CloudFormation, können Sie die Vorlage wiederverwenden, um Ihre Amazon Keyspaces Ressourcen einheitlich und wiederholt einzurichten. Sie beschreiben Ihre Ressourcen dann einmal und können die gleichen Ressourcen immer wieder bereitstellen.AWS-Konten

## Amazon Keyspaces undAWS CloudFormation Vorlagen

Um Ressourcen für Amazon Keyspaces bereitzustellen und zu konfigurieren, müssen Sie [AWS CloudFormation-Vorlagen](#) kennen und verstehen. Vorlagen sind formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation-Stacks bereitstellen möchten. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS CloudFormation Designer verwenden, der den Einstieg in die Arbeit mit AWS CloudFormation-Vorlagen erleichtert. Weitere Informationen [ist einAWS CloudFormation Designer?](#) im AWS CloudFormationBenutzerhandbuch.

Amazon Keyspaces unterstützt das Erstellen von Schlüsselräumen und Tabellen inAWS CloudFormation. Für die Tabellen, die Sie mithilfe vonAWS CloudFormation Vorlagen erstellen, können Sie das Schema, den Lese-/Schreibmodus und die bereitgestellten Durchsatzeinstellungen angeben. Weitere Informationen, einschließlich Beispiele für JSON- und YAML-Vorlagen für Schlüsselräume und Tabellen, ist der [Referenz zum Cassandra-Ressourcentyp](#) im AWS CloudFormation-Benutzerhandbuch.

## Weitere Informationen zu AWS CloudFormation

Weitere Informationen zu AWS CloudFormation finden Sie in den folgenden Ressourcen.



- [AWS CloudFormation](#)
- [AWS CloudFormation-Benutzerhandbuch](#)
- [AWS CloudFormation Benutzerhandbuch für die Befehlszeilenschnittstelle](#)

# Überwachung von Amazon Keyspaces (für Apache Cassandra)

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

- Amazon Keyspaces bietet ein vorkonfiguriertes Dashboard, in dem die Latenz und die Fehler zusammengefasst für alle Tabellen im Konto AWS Management Console angezeigt werden.
- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Mit maßgeschneiderten Dashboards können Sie Kennzahlen sammeln und verfolgen. Sie können beispielsweise eine Ausgangsbasis für die normale Leistung von Amazon Keyspaces in Ihrer Umgebung erstellen, indem Sie die Leistung zu verschiedenen Zeiten und unter verschiedenen Lastbedingungen messen. Speichern Sie bei der Überwachung von Amazon Keyspaces historische Überwachungsdaten, damit Sie sie mit aktuellen Leistungsdaten vergleichen, normale Leistungsmuster und Leistungsanomalien identifizieren und Methoden zur Behebung von Problemen entwickeln können. Um eine Ausgangsbasis festzulegen, sollten Sie mindestens nach Systemfehlern Ausschau halten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- CloudWatch Amazon-Alarme überwachen eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum und führen eine oder mehrere Aktionen auf der Grundlage des Werts der Metrik im Verhältnis zu einem bestimmten Schwellenwert über mehrere Zeiträume aus. Wenn Sie Amazon Keyspaces beispielsweise im Bereitstellungsmodus mit Application Auto Scaling verwenden, handelt es sich bei der Aktion um eine Benachrichtigung, die vom Amazon Simple Notification Service (Amazon SNS) gesendet wird, um eine Application Auto Scaling-Richtlinie auszuwerten.

CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Zustand befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter [Überwachung von Amazon Keyspaces mit Amazon CloudWatch](#).

- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien aus Amazon Keyspaces-Tabellen und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr

robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).

- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS-Konto-Kontos erfolgten, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon-S3-Bucket. Sie können die Benutzer und Konten, die AWS aufgerufen haben, identifizieren, sowie die Quell-IP-Adresse, von der diese Aufrufe stammen, und den Zeitpunkt der Aufrufe ermitteln. Weitere Informationen finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

Amazon EventBridge ist ein serverloser Event-Bus-Service, der es einfach macht, Ihre Anwendungen mit Daten aus einer Vielzahl von Quellen zu verbinden. EventBridge liefert einen Stream von Echtzeitdaten aus Ihren eigenen Anwendungen, oftware-as-a S-Service (SaaS) -Anwendungen und AWS Diensten und leitet diese Daten an Ziele wie Lambda weiter. Auf diese Weise können Sie Ereignisse überwachen, die in Services auftreten, und ereignisgesteuerte Architekturen erstellen. Weitere Informationen finden Sie im [EventBridge Amazon-Benutzerhandbuch](#).

## Themen

- [Überwachung von Amazon Keyspaces mit Amazon CloudWatch](#)
- [Protokollieren von Amazon Keyspaces-API-Aufrufen mit AWS CloudTrail](#)

# Überwachung von Amazon Keyspaces mit Amazon CloudWatch

Sie können Amazon Keyspaces mithilfe von Amazon überwachen. Amazon CloudWatch sammelt Rohdaten und verarbeitet sie zu lesbaren Metriken, die nahezu in Echtzeit ablaufen. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden.

Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

### Note

Um schnell mit einem vorkonfigurierten CloudWatch Dashboard zu beginnen, das allgemeine Kennzahlen für Amazon Keyspaces anzeigt, können Sie eine AWS CloudFormation Vorlage

verwenden, die unter verfügbar ist. <https://github.com/aws-samples/amazon-keyspaces-cloudwatch-cloudformation-templates>

## Themen

- [Wie verwende ich Amazon Keyspaces-Metriken?](#)
- [Amazon Keyspaces-Metriken und Dimensionen](#)
- [CloudWatch Alarme zur Überwachung von Amazon Keyspaces erstellen](#)

## Wie verwende ich Amazon Keyspaces-Metriken?

Die von Amazon Keyspaces gemeldeten Metriken liefern Informationen, die Sie auf unterschiedliche Weise analysieren können. In der folgenden Liste finden Sie einige häufige Verwendungszwecke für die Metriken. Es handelt sich dabei um Vorschläge für den Einstieg und nicht um eine umfassende Liste. Weitere Informationen zu Kennzahlen und Aufbewahrung finden Sie unter [Metriken](#).

| Wie kann ich ...?                                                                                                | Relevante Metriken                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Wie kann ich feststellen, ob Systemfehler aufgetreten sind?                                                      | Sie können <code>überwachenSystemErrors</code> , ob Anfragen zu einem Serverfehlercode geführt haben. In der Regel sollte diese Metrik gleich Null sein. Ist dies nicht der Fall, sollten Sie das genauer untersuchen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Wie kann ich die durchschnittliche bereitgestellte Lesekapazität mit der verbrauchten Lesekapazität vergleichen? | Um die durchschnittliche bereitgestellte Lesekapazität und die verbrauchte Lesekapazität zu überwachen <ol style="list-style-type: none"> <li>1. Stellen Sie den Zeitraum für <code>ConsumedReadCapacityUnits</code> und <code>ProvisionedReadCapacityUnits</code> auf das Intervall ein, das Sie überwachen möchten.</li> <li>2. Ändern Sie die Statistik für <code>ConsumedReadCapacityUnits</code> von <code>Average</code> bis <code>Sum</code>.</li> <li>3. Erstellen Sie einen neuen leeren Math-Ausdruck.</li> <li>4. Geben Sie im Abschnitt <code>Details</code> des neuen mathematischen Ausdrucks die ID von ein <b><code>ConsumedReadCapacityUnits</code></b> und dividieren Sie die Metrik durch die</li> </ol> |

| Wie kann ich ...?                                                                                                             | Relevante Metriken                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                               | <p>CloudWatch PERIOD-Funktion der Metrik (<code>metric_id/PERIOD (metric_id)</code>).</p> <p>5. Auswahl aufheben. <code>ConsumedReadCapacityUnits</code></p> <p>Sie können jetzt Ihre durchschnittlich verbrauchte Lesekapazität mit Ihrer bereitgestellten Kapazität vergleichen. Weitere Informationen zu grundlegenden arithmetischen Funktionen und zum Erstellen einer Zeitreihe finden Sie unter <a href="#">Metrische Mathematik verwenden</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <p>Wie kann ich die durchschnittliche bereitgestellte Schreibkapazität mit der verbrauchten Schreibkapazität vergleichen?</p> | <p>Um die durchschnittliche bereitgestellte Schreibkapazität und die verbrauchte Schreibkapazität zu überwachen</p> <ol style="list-style-type: none"> <li>1. Stellen Sie den Zeitraum für <code>ConsumedWriteCapacityUnits</code> und <code>ProvisionedWriteCapacityUnits</code> auf das Intervall ein, das Sie überwachen möchten.</li> <li>2. Ändern Sie die Statistik für <code>ConsumedWriteCapacityUnits</code> von <code>Average</code> bis <code>Sum</code>.</li> <li>3. Erstellen Sie einen neuen leeren Math-Ausdruck.</li> <li>4. Geben Sie im Abschnitt <code>Details</code> des neuen mathematischen Ausdrucks die ID von ein <b><code>ConsumedWriteCapacityUnits</code></b> und dividieren Sie die Metrik durch die CloudWatch PERIOD-Funktion der Metrik (<code>metric_id/PERIOD (metric_id)</code>).</li> <li>5. Auswahl aufheben. <code>ConsumedWriteCapacityUnits</code></li> </ol> <p>Sie können jetzt Ihre durchschnittlich verbrauchte Schreibkapazität mit Ihrer bereitgestellten Kapazität vergleichen. Weitere Informationen zu grundlegenden arithmetischen Funktionen und zum Erstellen einer Zeitreihe finden Sie unter <a href="#">Metrische Mathematik verwenden</a>.</p> |

## Amazon Keyspaces-Metriken und Dimensionen

Wenn Sie mit Amazon Keyspaces interagieren, werden die folgenden Metriken und Dimensionen an Amazon CloudWatch gesendet. Alle Metriken werden aggregiert und jede Minute gemeldet. Sie können die folgenden Verfahren verwenden, um die Metriken für Amazon Keyspaces anzuzeigen.

Um Metriken mit der CloudWatch Konsole anzuzeigen

Metriken werden zunächst nach dem Service-Namespace und anschließend nach den verschiedenen Dimensionskombinationen in den einzelnen Namespaces gruppiert.

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Ändern Sie, falls erforderlich, die Region. Wählen Sie in der Navigationsleiste die Region aus, in der sich Ihre AWS Ressourcen befinden. Weitere Informationen finden Sie unter [AWS -Service-Endpunkte](#).
3. Wählen Sie im Navigationsbereich Metriken aus.
4. Wählen Sie unter dem Tab Alle Metriken AWS/Cassandra .

So zeigen Sie Metriken mit der AWS CLI an

- Geben Sie als Eingabeaufforderung den folgenden Befehl ein.

```
aws cloudwatch list-metrics --namespace "AWS/Cassandra"
```

## Amazon Keyspaces-Metriken und Dimensionen

Die Metriken und Dimensionen, die Amazon Keyspaces an Amazon sendet, CloudWatch sind hier aufgeführt.

### Amazon-Keyspaces-Metriken

Amazon CloudWatch aggregiert Amazon Keyspaces-Metriken in Intervallen von einer Minute.

Nicht alle Statistiken, wie Average oder Sum sind für jede Metrik anwendbar. All diese Werte sind jedoch über die Amazon Keyspaces-Konsole oder über die CloudWatch Konsole oder AWS SDKs für alle Metriken verfügbar. AWS CLI In der folgenden Tabelle enthält jede Metrik eine Liste gültiger Statistiken, die für diese Metrik gelten.


| Metrik                                                 | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>AccountMaxTableLevelReads</code>                 | <p>Die maximale Anzahl von Lesekapazitätseinheiten, die von einer Tabelle eines Kontos verwendet werden können. Bei On-Demand-Tabellen begrenzt dieses Limit die maximale Anzahl von Leseanforderungseinheiten, die eine Tabelle verwenden kann.</p> <p>Einheiten: Count</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b>— Die maximale Anzahl von Lesekapazitätseinheiten, die von einer Tabelle des Kontos verwendet werden können.</li></ul>            |
| <code>AccountMaxTableLevelWrites</code>                | <p>Die maximale Anzahl von Schreibkapazitätseinheiten, die von einer Tabelle eines Kontos verwendet werden können. Bei On-Demand-Tabellen begrenzt dieses Limit die maximale Anzahl von Schreibenanforderungseinheiten, die eine Tabelle verwenden kann.</p> <p>Einheiten: Count</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b>— Die maximale Anzahl von Schreibkapazitätseinheiten, die von einer Tabelle des Kontos verwendet werden können.</li></ul> |
| <code>AccountProvisionedReadCapacityUtilization</code> | <p>Prozentsatz der bereitgestellten Lesekapazitätseinheiten, die vom Konto genutzt werden.</p> <p>Einheiten: Percent</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b> – Der maximale Prozentsatz der bereitgestellten Lesekapazitätseinheiten, die vom Konto genutzt werden.</li></ul>                                                                                                                                                                     |



| Metrik                                                 | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                        | <ul style="list-style-type: none"> <li>• <b>Minimum</b> – Der Mindestprozentsatz der bereitgestellten Lesekapazitätseinheiten, die vom Konto genutzt werden.</li> <li>• <b>Average</b> – Der durchschnittliche Prozentsatz der bereitgestellten Lesekapazitätseinheiten, die vom Konto genutzt werden. Die Metrik wird für Intervallen von fünf Minuten veröffentlicht. Wenn Sie die bereitgestellten Lesekapazitätseinheiten schnell anpassen, spiegelt diese Statistik möglicherweise nicht den tatsächlichen Durchschnitt wider.</li> </ul>                                                                                                                                                                                                                                                                                                        |
| <p>AccountProvisioned<br/>WriteCapacityUtilization</p> | <p>Prozentsatz der bereitgestellten Schreibkapazitätseinheiten, die vom Konto genutzt werden.</p> <p>Einheiten: Percent</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"> <li>• <b>Maximum</b> – Der maximale Prozentsatz der bereitgestellten Schreibkapazitätseinheiten, die vom Konto genutzt werden.</li> <li>• <b>Minimum</b> – Der minimale Prozentsatz der bereitgestellten Schreibkapazitätseinheiten, die vom Konto genutzt werden.</li> <li>• <b>Average</b> – Der durchschnittliche Prozentsatz der bereitgestellten Schreibkapazitätseinheiten, die vom Konto genutzt werden. Die Metrik wird für Intervallen von fünf Minuten veröffentlicht. Wenn Sie die bereitgestellten Schreibkapazitätseinheiten schnell anpassen, spiegelt diese Statistik möglicherweise nicht den tatsächlichen Durchschnitt wider.</li> </ul> |






| Metrik                                | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>BillableTableSizeInBytes</code> | <p>Die fakturierbare Größe der Tabelle in Byte. Es ist die Summe der kodierten Größe aller Zeilen in der Tabelle. Diese Metrik hilft Ihnen dabei, Ihre Tabellenspeicherkosten im Laufe der Zeit zu verfolgen.</p> <p>Einheiten: Bytes</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b>— Die maximale Speichergröße der Tabelle.</li><li>• <b>Minimum</b>— Die minimale Speichergröße der Tabelle.</li><li>• <b>Average</b>— Die durchschnittliche Speichergröße der Tabelle. Diese Metrik wird in Intervallen von 4 bis 6 Stunden berechnet.</li></ul> |

| Metrik                         | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConditionalCheckFailedRequests | <p>Die Anzahl der fehlgeschlagenen LWT-Schreibanforderungen (Lightweight Transaction). Mit den INSERT-, UPDATE- und DELETE-Operationen können Sie eine logische Bedingung angeben, die als wahr ausgewertet werden muss, bevor die Operation fortgesetzt werden kann. Wenn diese Bedingung als falsch ausgewertet wird, wird <code>ConditionalCheckFailedRequests</code> um eins erhöht. Bedingungsprüfungen, die den Wert falsch ergeben, verbrauchen Schreibkapazitätseinheiten auf der Grundlage der Zeilengröße. Weitere Informationen finden Sie unter <a href="#">the section called “Leichte Transaktionen”</a>.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• Minimum</li><li>• Maximum</li><li>• Average</li><li>• SampleCount</li><li>• Sum</li></ul> |

| Metrik                    | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConsumedReadCapacityUnits | <p>Die Anzahl der im angegebenen Zeitraum verbrauchten Lesekapazitätseinheiten. Weitere Informationen finden Sie unter <a href="#">Lese-/Schreibkapazitätsmodus</a>.</p> <div data-bbox="688 401 1507 1285" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Um Ihre durchschnittliche Durchsatzauslastung pro Sekunde zu ermitteln, verwenden Sie die Sum Statistiken, um den verbrauchten Durchsatz für den Zeitraum von einer Minute zu berechnen. Teilen Sie dann die Summe durch die Anzahl der Sekunden in einer Minute (60), um den Durchschnitt ConsumedReadCapacityUnits pro Sekunde zu berechnen (wobei zu berücksichtigen ist, dass dieser Durchschnitt keine großen, sondern kurze Spitzen der Leseaktivität hervorhebt, die während dieser Minute aufgetreten sind). Weitere Informationen zum Vergleich der durchschnittlich verbrauchten Lesekapazität mit der bereitgestellten Lesekapazität finden Sie unter <a href="#">the section called “Verwenden von - Metriken”</a></p></div> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Minimum</b>— Die Mindestanzahl an Lesekapazitätseinheiten, die von einer einzelnen Anforderung an die Tabelle verbraucht werden.</li><li>• <b>Maximum</b>— Die maximale Anzahl von Lesekapazitätseinheiten, die von einer einzelnen Anfrage an die Tabelle verbraucht werden.</li></ul> |

| Metrik | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <ul style="list-style-type: none"><li data-bbox="688 212 1442 289">• <code>Average</code> – Die durchschnittliche Lesekapazität pro Anforderung.</li></ul> <div data-bbox="716 331 1507 600"><p> <b>Note</b></p><p>Der <code>Average</code>-Wert wird von Perioden der Inaktivität beeinflusst, in denen der Stichprobenwert Null ist.</p></div> <ul style="list-style-type: none"><li data-bbox="688 621 1490 751">• <code>Sum</code> – Die verbrauchten Einheiten der gesamten Lesekapazität. Die nützlichste Statistik für diese Metrik ist <code>ConsumedReadCapacityUnits</code> .</li><li data-bbox="688 772 1507 898">• <code>SampleCount</code> — Die Anzahl der Anfragen an Amazon Keyspaces, auch wenn keine Lesekapazität verbraucht wurde.</li></ul> <div data-bbox="716 940 1507 1209"><p> <b>Note</b></p><p>Der <code>SampleCount</code> -Wert wird von Perioden der Inaktivität beeinflusst, in denen der Stichprobenwert Null ist.</p></div> |

| Metrik                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConsumedWriteCapacityUnits | <p>Die Anzahl der Schreibkapazitätseinheiten, die im angegebenen Zeitraum verbraucht wurden. Sie können die gesamte verbrauchte Schreibkapazität für eine Tabelle abrufen. Weitere Informationen finden Sie unter <a href="#">Lese-/Schreibkapazitätsmodus</a>.</p> <div data-bbox="688 495 1507 1381" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Um Ihre durchschnittliche Durchsatzauslastung pro Sekunde zu ermitteln, verwenden Sie die Sum Statistik, um den verbrauchten Durchsatz für den Zeitraum von einer Minute zu berechnen. Teilen Sie dann die Summe durch die Anzahl der Sekunden in einer Minute (60), um den Durchschnitt ConsumedWriteCapacityUnits pro Sekunde zu berechnen (wobei zu berücksichtigen ist, dass dieser Durchschnitt keine großen, sondern nur kurze Spitzen der Schreibaktivität hervorhebt, die während dieser Minute aufgetreten sind). Weitere Informationen zum Vergleich der durchschnittlich verbrauchten Schreibkapazität mit der bereitgestellten Schreibkapazität finden Sie unter <a href="#">the section called “Verwenden von -Metriken”</a></p></div> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Minimum</b>— Die Mindestanzahl von Schreibkapazitätseinheiten, die von einer einzelnen Anforderung an die Tabelle verbraucht werden.</li></ul> |

| Metrik | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | <ul style="list-style-type: none"><li>• <b>Maximum</b>— Die maximale Anzahl von Schreibkapazitätseinheiten, die von einer einzelnen Anfrage an die Tabelle verbraucht werden.</li><li>• <b>Average</b> – Die durchschnittliche Schreibkapazität pro Anforderung.</li></ul> <div data-bbox="716 485 1507 751" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Der Average-Wert wird von Perioden der Inaktivität beeinflusst, in denen der Stichprobenwert Null ist.</p></div> <ul style="list-style-type: none"><li>• <b>Sum</b> – Die verbrauchten Schreibkapazitätseinheiten. Die nützlichste Statistik für diese Metrik ist <code>ConsumedWriteCapacityUnits</code>.</li><li>• <b>SampleCount</b> — Die Anzahl der Anfragen an Amazon Keyspaces, auch wenn keine Schreibkapazität verbraucht wurde.</li></ul> <div data-bbox="716 1094 1507 1360" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Der SampleCount -Wert wird von Perioden der Inaktivität beeinflusst, in denen der Stichprobenwert Null ist.</p></div> |

| Metrik                                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MaxProvisionedTableReadCapacityUtilization | <p>Der Prozentsatz der bereitgestellten Lesekapazitätseinheiten, der von der Tabelle mit der höchsten bereitgestellten Lesekapazität eines Kontos genutzt wird.</p> <p>Einheiten: Percent</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b> – : Der maximale Prozentsatz der bereitgestellten Lesekapazitätseinheiten, die von der höchsten bereitgestellten Lesetabelle des Kontos genutzt werden.</li><li>• <b>Minimum</b> – Der Mindestprozentsatz der bereitgestellten Lesekapazitätseinheiten, die von der höchsten bereitgestellten Lesetabelle des Kontos genutzt werden.</li><li>• <b>Average</b> – Der durchschnittliche Prozentsatz der bereitgestellten Lesekapazitätseinheiten, die von der höchsten bereitgestellten Lesetabelle des Kontos genutzt werden. Die Metrik wird für Intervallen von fünf Minuten veröffentlicht. Wenn Sie die bereitgestellten Lesekapazitätseinheiten schnell anpassen, spiegelt diese Statistik möglicherweise nicht den tatsächlichen Durchschnitt wider.</li></ul> |

| Metrik                                      | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MaxProvisionedTableWriteCapacityUtilization | <p data-bbox="686 226 1450 359">Der Prozentsatz der bereitgestellten Schreibkapazität, der von der Tabelle mit der höchsten bereitgestellten Schreibkapazität eines Kontos genutzt wird.</p> <p data-bbox="686 405 971 436">Einheiten: Percent</p> <p data-bbox="686 483 954 514">Gültige Statistiken:</p> <ul data-bbox="686 560 1500 1329" style="list-style-type: none"><li data-bbox="686 560 1500 741">• <b>Maximum</b>— Der maximale Prozentsatz der bereitgestellten Schreibkapazitätseinheiten, der von der Tabelle mit der höchsten bereitgestellten Schreibkapazität des Kontos genutzt wird.</li><li data-bbox="686 762 1500 942">• <b>Minimum</b>— Der Mindestprozentsatz der bereitgestellten Schreibkapazitätseinheiten, der von der Tabelle mit der höchsten bereitgestellten Schreibkapazität des Kontos genutzt wird.</li><li data-bbox="686 963 1500 1329">• <b>Average</b>— Der durchschnittliche Prozentsatz der bereitgestellten Schreibkapazitätseinheiten, der von der Tabelle mit der höchsten bereitgestellten Schreibkapazität des Kontos genutzt wird. Die Metrik wird für Intervallen von fünf Minuten veröffentlicht. Wenn Sie die bereitgestellten Schreibkapazitätseinheiten schnell anpassen, spiegelt diese Statistik möglicherweise nicht den tatsächlichen Durchschnitt wider.</li></ul> |




| Metrik                           | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PerConnectionRequestRateExceeded | <p>Anfragen an Amazon Keyspaces, die das Kontingent für die Anforderungsrate pro Verbindung überschreiten. Jede Client-Verbindung zu Amazon Keyspaces kann bis zu 3000 CQL-Anfragen pro Sekunde unterstützen. Clients können mehrere Verbindungen herstellen, um den Durchsatz zu erhöhen.</p> <p>Wenn Sie die regionsübergreifende Replikation verwenden, trägt jeder replizierte Schreibvorgang ebenfalls zu diesem Kontingent bei. Als bewährte Methode empfehlen wir, die Anzahl der Verbindungen zu Ihren Tabellen zu erhöhen, um Fehler zu vermeiden <code>PerConnectionRequestRateExceeded</code>. Die Anzahl der Verbindungen, die Sie in Amazon Keyspaces haben können, ist unbegrenzt.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName, Operation</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• SampleCount</li><li>• Sum</li></ul> |

| Metrik                       | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ProvisionedReadCapacityUnits | <p>Die Anzahl der bereitgestellten Lesekapazitätseinheiten für eine Tabelle.</p> <p>Die <code>TableName</code> Dimension gibt den <code>ProvisionedReadCapacityUnits</code> für die Tabelle zurück.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Minimum</b> – Die niedrigste Einstellung für bereitgestellte Lesekapazität. Wenn Sie <code>ALTER TABLE</code> verwenden, um die Lesekapazität zu erhöhen, zeigt diese Metrik den niedrigsten Wert der bereitgestellten <code>ReadCapacityUnits</code> während dieses Zeitraums an.</li><li>• <b>Maximum</b> – Die höchste Einstellung für bereitgestellte Lesekapazität. Wenn Sie <code>ALTER TABLE</code> verwenden, um die Lesekapazität zu verringern, zeigt diese Metrik den höchsten Wert der bereitgestellten <code>ReadCapacityUnits</code> während dieses Zeitraums an.</li><li>• <b>Average</b> – Die durchschnittliche bereitgestellte Lesekapazität. Die <code>ProvisionedReadCapacityUnits</code>-Metrik wird in Intervallen von fünf Minuten veröffentlicht. Wenn Sie die bereitgestellten Lesekapazitätseinheiten schnell anpassen, spiegelt diese Statistik möglicherweise nicht den tatsächlichen Durchschnitt wider.</li></ul> |

| Metrik                                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ProvisionedWriteCapacityUnits</code> | <p>Die Anzahl der bereitgestellten Schreibkapazitätseinheiten für eine Tabelle.</p> <p>Die <code>TableName</code> Dimension gibt den <code>ProvisionedWriteCapacityUnits</code> für die Tabelle zurück.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <b>Minimum</b> – Die niedrigste Einstellung für bereitgestellte Schreibkapazität. Wenn Sie <code>ALTER TABLE</code> verwenden, um die Schreibkapazität zu erhöhen, zeigt diese Metrik den niedrigsten Wert der bereitgestellten <code>WriteCapacityUnits</code> während dieses Zeitraums an.</li><li>• <b>Maximum</b> – Die höchste Einstellung für bereitgestellte Schreibkapazität. Wenn Sie <code>ALTER TABLE</code> verwenden, um die Schreibkapazität zu verringern, zeigt diese Metrik den höchsten Wert der bereitgestellten <code>WriteCapacityUnits</code> während dieses Zeitraums an.</li><li>• <b>Average</b> – Die durchschnittliche bereitgestellte Schreibkapazität. Die <code>ProvisionedWriteCapacityUnits</code>-Metrik wird in Intervallen von fünf Minuten veröffentlicht. Wenn Sie die bereitgestellten Schreibkapazitätseinheiten schnell anpassen, spiegelt diese Statistik möglicherweise nicht den tatsächlichen Durchschnitt wider.</li></ul> |

| Metrik                          | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ReadThrottleEvents</code> | <p>Anfragen an Amazon Keyspaces, die die bereitgestellte Lesekapazität für eine Tabelle oder Kontingente auf Kontoebene, Quoten pro Anfrage pro Verbindung oder Kontingente auf Partitionsebene überschreiten.</p> <p>Einheiten: Count</p> <p>Maße: <code>Keyspace</code>, <code>TableName</code>, <code>Operation</code></p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <code>SampleCount</code></li><li>• <code>Sum</code></li></ul>                                                                                                  |
| <code>ReplicationLatency</code> | <p>Diese Metrik gilt nur für Schlüsselräume mit mehreren Regionen und misst die Zeitupdates, <code>inserts</code> die für die Replikation oder <code>deletes</code> von einer Replikattabelle zu einer anderen Replikattabelle in einem Schlüsselraum mit mehreren Regionen benötigt wurde.</p> <p>Einheiten: Millisecond</p> <p>Maße: <code>TableName</code>, <code>ReceivingRegion</code></p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <code>Average</code></li><li>• <code>Maximum</code></li><li>• <code>Minimum</code></li></ul> |

| Metrik                    | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReturnedItemCountBySelect | <p>Die Anzahl der Zeilen, die von SELECT mehrzeiligen Abfragen während des angegebenen Zeitraums zurückgegeben wurden. Mehrzeilige SELECT Abfragen sind Abfragen, die nicht den vollqualifizierten Primärschlüssel enthalten, wie z. B. vollständige Tabellenscans und Bereichsabfragen.</p> <p>Die Anzahl der zurückgegebenen Zeilen entspricht nicht unbedingt der Anzahl der Zeilen, die ausgewertet wurden. Nehmen wir zum Beispiel an, Sie haben für eine Tabelle <code>SELECT *</code> mit <code>ALLOW FILTERING</code> 100 Zeilen ein mit angefordert, aber eine <code>WHERE</code> Klausel angegeben, die die Ergebnisse so einschränkt, dass nur 15 Zeilen zurückgegeben werden. In diesem Fall <code>SELECT</code> würde die Antwort von <code>a 100</code> und <code>a ScanCount</code> <code>Count</code> von 15 zurückgegebenen Zeilen enthalten.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName, Operation</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• Minimum</li><li>• Maximum</li><li>• Average</li><li>• SampleCount</li><li>• Sum</li></ul> |

| Metrik                                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| StoragePartitionThroughputCapacityExceeded | <p>Anfragen an eine Amazon Keyspaces-Speicherpartition, die die Durchsatzkapazität der Partition überschreiten. Amazon Keyspaces-Speicherpartitionen können bis zu 1000 WCU/WRU pro Sekunde und 3000 RCU/RRU pro Sekunde unterstützen. Wir empfehlen, Ihr Datenmodell zu überprüfen, um den Lese-/Schreibverkehr auf mehrere Partitionen zu verteilen, um diese Ausnahmen zu vermeiden.</p> <div data-bbox="686 636 1508 905" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Logische Amazon Keyspaces-Partitionen können sich über mehrere Speicherpartitionen erstrecken und sind praktisch unbegrenzt groß.</p> </div> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName, Operation</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"> <li>• SampleCount</li> <li>• Sum</li> </ul> |
| SuccessfulRequestCount                     | <p>Die Anzahl der erfolgreichen Anfragen, die im angegebenen Zeitraum verarbeitet wurden.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName, Operation</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"> <li>• SampleCount</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Metrik                   | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SuccessfulRequestLatency | <p>Die erfolgreichen Anfragen an Amazon Keyspaces während des angegebenen Zeitraums. SuccessfulRequestLatency kann zwei verschiedene Arten von Informationen bereitstellen:</p> <ul style="list-style-type: none"><li>• Die verstrichene Zeit für erfolgreiche Anfragen (Minimum, Maximum, Sum oder Average).</li><li>• Die Anzahl erfolgreicher Anforderungen (SampleCount).</li></ul> <p>SuccessfulRequestLatency spiegelt nur Aktivitäten innerhalb von Amazon Keyspaces wider und berücksichtigt weder Netzwerklatenz noch clientseitige Aktivitäten.</p> <p>Einheiten: Milliseconds</p> <p>Maße: Keyspace, TableName, Operation</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• Minimum</li><li>• Maximum</li><li>• Average</li><li>• SampleCount</li></ul> |

| Metrik                      | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SystemErrors                | <p>Die Anfragen an Amazon Keyspaces, die <code>ServerError</code> während des angegebenen Zeitraums eine generieren. A weist <code>ServerError</code> normalerweise auf einen internen Servicefehler hin.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName, Operation</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"> <li>• Sum</li> <li>• SampleCount</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| SystemReconciliationDeletes | <p>Die Einheiten, die zum Löschen von als veraltet eingestuft Daten verwendet werden, wenn clientseitige Zeitstempel aktiviert sind. Jede <code>SystemReconciliationDelete</code> bietet ausreichend Kapazität, um bis zu 1 KB Daten pro Zeile zu löschen oder zu aktualisieren. Um beispielsweise eine Zeile zu aktualisieren, die 2,5 KB an Daten speichert, und um eine oder mehrere Spalten in der Zeile gleichzeitig zu löschen, sind 3 <code>SystemReconciliationDeletes</code> erforderlich. Oder um eine ganze Zeile zu löschen, die 3,5 KB an Daten enthält, sind 4 <code>SystemReconciliationDeletes</code> erforderlich.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"> <li>• Sum— Die Gesamtzahl der in einem Zeitraum <code>SystemReconciliationDeletes</code> verbrauchten.</li> </ul> |



| Metrik     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TTLDeletes | <p>Die Einheiten, die zum Löschen oder Aktualisieren von Daten nacheinander mithilfe von Time to Live (TTL) verbraucht wurden. Jede TTLDelete bietet ausreichend Kapazität, um bis zu 1 KB Daten pro Zeile zu löschen oder zu aktualisieren. Um beispielsweise eine Zeile zu aktualisieren, die 2,5 KB an Daten speichert, und um eine oder mehrere Spalten innerhalb der Zeile gleichzeitig zu löschen, sind 3 TTL-Löschungen erforderlich. Oder um eine ganze Zeile zu löschen, die 3,5 KB an Daten enthält, sind 4 TTL-Löschungen erforderlich.</p> <p>Einheiten: Count</p> <p>Maße: Keyspace, TableName</p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• Sum— Die Gesamtzahl der in einem Zeitraum TTLDeletes verbrauchten.</li></ul> |

| Metrik                           | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>UserErrors</code>          | <p>Anfragen an Amazon Keyspaces, die während des angegebenen Zeitraums einen <code>InvalidRequest</code> Fehler generieren. Ein <code>InvalidRequest</code> Fehler weist normalerweise auf einen clientseitigen Fehler hin, z. B. eine ungültige Kombination von Parametern, einen Versuch, eine nicht existierende Tabelle zu aktualisieren, oder eine falsche Anforderungssignatur.</p> <p><code>UserErrors</code> stellt die Summe der ungültigen Anfragen für die aktuelle AWS-Region und die aktuelle Anfrage dar.<br/>AWS-Konto</p> <p>Einheiten: <code>Count</code></p> <p>Maße: <code>Keyspace</code>, <code>TableName</code>, <code>Operation</code></p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <code>Sum</code></li><li>• <code>SampleCount</code></li></ul> |
| <code>WriteThrottleEvents</code> | <p>Anfragen an Amazon Keyspaces, die die bereitgestellte Schreibkapazität für eine Tabelle oder Kontingente auf Kontoebene, Quoten pro Anfrage pro Verbindung oder Kontingente auf Partitionsebene überschreiten.</p> <p>Einheiten: <code>Count</code></p> <p>Maße: <code>Keyspace</code>, <code>TableName</code>, <code>Operation</code></p> <p>Gültige Statistiken:</p> <ul style="list-style-type: none"><li>• <code>SampleCount</code></li><li>• <code>Sum</code></li></ul>                                                                                                                                                                                                                                                                                                                     |

## Dimensionen für Amazon Keyspaces-Metriken

Die Metriken für Amazon Keyspaces werden anhand der Werte für das Konto, den Tabellennamen oder den Vorgang qualifiziert. Sie können die CloudWatch Konsole verwenden, um Amazon Keyspaces-Daten für jede der Dimensionen in der folgenden Tabelle abzurufen.

| Dimension | Beschreibung                                                                                                                                                                                                                                                                              |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keyspace  | Diese Dimension beschränkt die Daten auf einen bestimmten Schlüsselraum. Dieser Wert kann ein beliebiger Schlüsselraum in der aktuellen Region und in der aktuellen Region sein. AWS-Konto                                                                                                |
| Operation | Diese Dimension beschränkt die Daten auf eine der Amazon Keyspaces-CQL-Operationen, wie INSERT z. B. Oder-Operationen. SELECT                                                                                                                                                             |
| TableName | Diese Dimension schränkt die Daten auf einen spezifische Tabelle ein. Dieser Wert kann ein beliebiger Tabellename in der aktuellen Region und in der aktuellen Region sein. AWS-Konto Wenn der Tabellename innerhalb des Kontos nicht eindeutig ist, müssen Sie ihn auch angebenKeyspace. |

## CloudWatch Alarme zur Überwachung von Amazon Keyspaces erstellen

Sie können einen CloudWatch Amazon-Alarm für Amazon Keyspaces erstellen, der eine Amazon Simple Notification Service (Amazon SNS) -Nachricht sendet, wenn sich der Status des Alarms ändert. Ein Alarm überwacht eine Metrik über einen bestimmten, von Ihnen definierten Zeitraum. Der Alarm führt eine oder mehrere Aktionen durch, basierend auf dem Wert der Metrik im Vergleich zu einem bestimmten Schwellenwert in einer Reihe von Zeiträumen. Die Aktion ist eine Benachrichtigung, die an ein Amazon SNS SNS-Thema oder eine Application Auto Scaling Scaling-Richtlinie gesendet wird.

Wenn Sie Amazon Keyspaces im Bereitstellungsmodus mit Application Auto Scaling verwenden, erstellt der Service in Ihrem Namen zwei CloudWatch Alarmpaare. Jedes Paar steht für Ihre Ober- und Untergrenzen für bereitgestellte und verbrauchte Durchsatzeinstellungen. Diese CloudWatch Alarme werden ausgelöst, wenn die tatsächliche Auslastung der Tabelle über einen längeren

Zeitraum von Ihrer Zielauslastung abweicht. Weitere Informationen zu CloudWatch Alarmen, die von Application Auto Scaling erstellt wurden, finden Sie unter [the section called “So funktioniert die automatische Skalierung von Amazon Keyspace”](#).

Alarme rufen nur Aktionen für anhaltende Statusänderungen auf. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Zustand befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein.

Weitere Informationen zum Erstellen von CloudWatch Alarmen finden Sie unter [Verwenden von CloudWatch Amazon-Alarmen](#) im CloudWatch Amazon-Benutzerhandbuch.

## Protokollieren von Amazon Keyspaces-API-Aufrufen mit AWS CloudTrail

Amazon Keyspaces ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS Dienstes in Amazon Keyspaces bereitstellt. CloudTrail erfasst Data Definition Language (DDL) -API-Aufrufe und Data Manipulation Language (DML) -API-Aufrufe für Amazon Keyspaces als Ereignisse. Zu den aufgezeichneten Aufrufen gehören Aufrufe von der Amazon Keyspaces-Konsole und programmatische Aufrufe der Amazon Keyspaces-API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon Simple Storage Service (Amazon S3) -Bucket aktivieren, einschließlich Ereignissen für Amazon Keyspaces.

Wenn Sie keinen Trail konfigurieren, können Sie die zuletzt unterstützten Ereignisse trotzdem auf der CloudTrail Konsole im Event-Verlauf einsehen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon Keyspaces gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

### Themen

- [Konfiguration von Amazon Keyspaces-Protokolldateieinträgen in CloudTrail](#)
- [Informationen zur Amazon Keyspaces-Datendefinitionssprache \(DDL\) in CloudTrail](#)
- [Informationen zur Amazon Keyspaces Data Manipulation Language \(DML\) in CloudTrail](#)

- [Grundlegendes zu Amazon Keyspaces-Protokolldateieinträgen](#)

## Konfiguration von Amazon Keyspaces-Protokolldateieinträgen in CloudTrail

Jede angemeldete Amazon Keyspaces-API-Aktion CloudTrail enthält Anforderungsparameter, die in der CQL-Abfragesprache ausgedrückt werden. Weitere Informationen hierzu finden Sie unter [CQL-Sprachreferenz](#).

Sie können in Ihrem AWS-Konto die neusten Ereignisse anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich Veranstaltungen für Amazon Keyspaces, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren.

Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail -Benutzerhandbuch:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#)
- [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem [CloudTrail UserIdentity-Element](#).

## Informationen zur Amazon Keyspaces-Datendefinitionssprache (DDL) in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn eine DDL-Aktivität in Amazon Keyspaces auftritt, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen im CloudTrail Eventverlauf automatisch als Ereignis aufgezeichnet. Die folgende Tabelle zeigt die DDL-Anweisungen, die für Amazon Keyspaces protokolliert werden.

| CloudTrail eventName | Statement | CQL-Aktion      | AWS SDK-Aktion                                  |
|----------------------|-----------|-----------------|-------------------------------------------------|
| CreateKeyspace       | DDL       | CREATE KEYSPACE | CreateKeyspace                                  |
| DropKeyspace         | DDL       | DROP KEYSPACE   | DeleteKeyspace                                  |
| CreateTable          | DDL       | CREATE TABLE    | CreateTable                                     |
| DropTable            | DDL       | DROP TABLE      | DeleteTable                                     |
| AlterTable           | DDL       | ALTER TABLE     | UpdateTable ,<br>TagResource ,<br>UntagResource |

## Informationen zur Amazon Keyspaces Data Manipulation Language (DML) in CloudTrail

Um die Protokollierung von Amazon Keyspaces-DML-Anweisungen mit zu aktivieren CloudTrail, müssen Sie zuerst die Protokollierung der API-Aktivität auf der Datenebene aktivieren. CloudTrail Sie können mit der Protokollierung von Amazon Keyspaces-DML-Ereignissen in neuen oder bestehenden Trails beginnen, indem Sie festlegen, dass Aktivitäten für den Datenereignistyp Cassandra-Tabelle über die CloudTrail Konsole protokolliert werden sollen, oder indem AWS::Cassandra::Table Sie den resources.type Wert auf AWS CLI- oder CloudTrail API-Operationen setzen. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen](#).

Die folgende Tabelle zeigt die Datenereignisse, die von for protokolliert werden. CloudTrail  
Cassandra table

| CloudTrail eventName | Statement | CQL-Aktion | AWS SDK-Aktion                                                        |
|----------------------|-----------|------------|-----------------------------------------------------------------------|
| Select               | DML       | SELECT     | GetKeyspace, GetTable, ListKeyspaces, ListTables, ListTagsForResource |
| Einfügen             | DML       | INSERT     | keine AWS SDK-Aktionen verfügbar                                      |
| Aktualisierung       | DML       | UPDATE     | keine AWS SDK-Aktionen verfügbar                                      |
| Löschen              | DML       | DELETE     | keine AWS SDK-Aktionen verfügbar                                      |

## Grundlegendes zu Amazon Keyspaces-Protokolldateieinträgen

CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die DropTable Aktionen CreateKeyspace, DropKeyspaceCreateTable, und demonstriert:

```
{
 "Records": [
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
```

```

 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 },
 "eventTime": "2020-01-15T18:53:04Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "CreateKeyspace",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "rawQuery": "\n\tCREATE KEYSPACE \"mykeyspace\"\n\tWITH\n\t\tREPLICATION =
{'class': 'SingleRegionStrategy'}\n\t\t",
 "keyspaceName": "mykeyspace"
 },
 "responseElements": null,
 "requestID": "bfa3e75d-bf4d-4fc0-be5e-89d15850eb41",
 "eventID": "d25beae8-f611-4229-877a-921557a07bb9",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Keyspace",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "recipientAccountId": "111122223333",
 "managementEvent": true,
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",

```



```

 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 },
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 }
 }
 },
 "eventTime": "2020-01-15T19:28:39Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "DropKeyspace",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "rawQuery": "DROP KEYSPACE \"mykeyspace\"",
 "keyspaceName": "mykeyspace"
 },
 "responseElements": null,
 "requestID": "66f3d86a-56ae-4c29-b46f-abcd489ed86b",
 "eventID": "e5aebec-e1dd-41e3-a515-84fe6aaabd7b",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Keyspace",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/"
 }
]
}

```

```

 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "recipientAccountId": "111122223333",
 "managementEvent": true,
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 },
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 }
 }
 },
 "eventTime": "2020-01-15T18:55:24Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "CreateTable",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "rawQuery": "\n\tCREATE TABLE \"mykeyspace\".\"mytable\"(\n\t\t\"ID\" int,\n\t\t\"username\" text,\n\t\t\"email\" text,\n\t\t\"post_type\" text,\n\t\tPRIMARY\n\t\tKEY((\"ID\", \"username\", \"email\")))",
 }
}

```

```

 "keyspaceName": "mykeyspace",
 "tableName": "mytable"
 },
 "responseElements": null,
 "requestID": "5f845963-70ea-4988-8a7a-2e66d061aacb",
 "eventID": "fe0dbd2b-7b34-4675-a30c-740f9d8d73f9",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "recipientAccountId": "111122223333",
 "managementEvent": true,
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 },
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 }
 }
 }
}

```

```

 }
 }
},
"eventTime": "2020-01-15T19:27:59Z",
"eventSource": "cassandra.amazonaws.com",
"eventName": "DropTable",
"awsRegion": "us-east-1",
"sourceIPAddress": "10.24.34.01",
"userAgent": "Cassandra Client/ProtocolV4",
"requestParameters": {
 "rawQuery": "DROP TABLE \"mykeyspace\".\"mytable\"\"",
 "keyspaceName": "mykeyspace",
 "tableName": "mytable"
},
"responseElements": null,
"requestID": "025501b0-3582-437e-9d18-8939e9ef262f",
"eventID": "1a5cbcdc-4e38-4889-8475-3eab98de0ffd",
"readOnly": false,
"resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable"
 }
],
"eventType": "AwsApiCall",
"apiVersion": "3.4.4",
"recipientAccountId": "111122223333",
"managementEvent": true,
"eventCategory": "Management",
"tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
}
]
}

```

Die folgende Protokolldatei zeigt ein Beispiel für eine SELECT Anweisung.

```

{
 "eventVersion": "1.09",

```

```
"userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice"
},
"eventTime": "2023-11-17T10:38:04Z",
"eventSource": "cassandra.amazonaws.com",
"eventName": "Select",
"awsRegion": "us-east-1",
"sourceIPAddress": "10.24.34.01",
"userAgent": "Cassandra Client/ProtocolV4",
"requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "conditions": [
 "pk = *(Redacted)",
 "ck < 3*(Redacted)0",
 "region = 't*(Redacted)t'"
],
 "select": [
 "pk",
 "ck",
 "region"
],
 "allowFiltering": true
},
"responseElements": null,
"requestID": "6d83bbf0-a3d0-4d49-b1d9-e31779a28628",
"eventID": "e00552d3-34e9-4092-931a-912c4e08ba17",
"readOnly": true,
"resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/
table/my_table"
 }
],
"eventType": "AwsApiCall",
"apiVersion": "3.4.4",
"managementEvent": false,
"recipientAccountId": "111122223333",
```

```

"eventCategory": "Data",
"tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
}
}

```

Die folgende Protokolldatei zeigt ein Beispiel für eine INSERT Anweisung.

```

{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice"
 },
 "eventTime": "2023-12-01T22:11:43Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Insert",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "primaryKeys": {
 "pk": "**(Redacted)",
 "ck": "1**(Redacted)8"
 },
 "columnNames": [
 "pk",
 "ck",
 "region"
],
 "updateParameters": {
 "TTL": "2**(Redacted)0"
 }
 },
 "responseElements": null,
}

```

```

"requestID": "edf8af47-2f87-4432-864d-a960ac35e471",
"eventID": "81b56a1c-9bdd-4c92-bb8e-92776b5a3bf1",
"readOnly": false,
"resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/
my_table"
 }
],
"eventType": "AwsApiCall",
"apiVersion": "3.4.4",
"managementEvent": false,
"recipientAccountId": "111122223333",
"eventCategory": "Data",
"tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
}
}

```

Die folgende Protokolldatei zeigt ein Beispiel für eine UPDATE Anweisung.

```

{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice"
 },
 "eventTime": "2023-12-01T22:11:43Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Update",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 }
}

```

```

 "primaryKeys": {
 "pk": "'t**(Redacted)t'",
 "ck": "'s**(Redacted)g'"
 },
 "assignmentColumnNames": [
 "nonkey"
],
 "conditions": [
 "nonkey < 1**(Redacted)7"
]
 },
 "responseElements": null,
 "requestID": "edf8af47-2f87-4432-864d-a960ac35e471",
 "eventID": "81b56a1c-9bdd-4c92-bb8e-92776b5a3bf1",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/
my_table"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "managementEvent": false,
 "recipientAccountId": "111122223333",
 "eventCategory": "Data",
 "tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 }
}

```

Die folgende Protokolldatei zeigt ein Beispiel für eine DELETE Anweisung.

```

{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",

```



```
 "accountId": "111122223333",
 "userName": "alice",
 },
 "eventTime": "2023-10-23T13:59:05Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Delete",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "primaryKeys": {
 "pk": "**(Redacted)",
 "ck": "**(Redacted)"
 },
 "conditions": [],
 "deleteColumnNames": [
 "m",
 "s"
],
 "updateParameters": {}
 },
 "responseElements": null,
 "requestID": "3d45e63b-c0c8-48e2-bc64-31afc5b4f49d",
 "eventID": "499da055-c642-4762-8775-d91757f06512",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/
my_table"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "managementEvent": false,
 "recipientAccountId": "111122223333",
 "eventCategory": "Data",
 "tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
```

```
}
}
```

# Sicherheit in Amazon Keyspaces (für Apache Cassandra)

Die Sicherheit in der Cloud hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und als 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. Die Wirksamkeit unserer Sicherheitsfunktionen wird regelmäßig von externen Prüfern im Rahmen des [AWS-Compliance-Programms getestet und überprüft](#). Weitere Informationen zu den Compliance-Programmen, die für Amazon Keyspaces gelten, finden Sie unter [AWSServices im Umfang nach Compliance-Programmen](#).
- 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.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amazon Keyspaces anwenden können. In den folgenden Themen erfahren Sie, wie Sie Amazon Keyspaces konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen helfen können, Ihre Amazon Keyspaces-Ressourcen zu überwachen und zu sichern.

## Themen

- [Datenschutz in Amazon Keyspaces](#)
- [AWS Identity and Access Management für Amazon Keyspaces](#)
- [Konformitätsprüfung für Amazon Keyspaces \(für Apache Cassandra\)](#)
- [Ausfallsicherheit und Notfallwiederherstellung](#)
- [Infrastruktursicherheit in Amazon Keyspaces](#)
- [Konfigurations- und Schwachstellenanalyse für Amazon Keyspaces](#)
- [Best Practices für die Sicherheit in Amazon Keyspaces](#)

# Datenschutz in Amazon Keyspaces

Das [Modell der AWS gemeinsamen Verantwortung](#) gilt für den Datenschutz in Amazon Keyspaces (für Apache Cassandra). Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, 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 zu 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-Endpoint. Weitere Informationen über verfügbare FIPS-Endpoints 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 Keyspaces oder anderen AWS-Services über die Konsole, AWS CLI, API 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

- [Verschlüsselung](#)
- [Verschlüsselung während der Übertragung in Amazon Keyspaces](#)
- [Datenschutz im Netzwerkverkehr in Amazon Keyspaces](#)

## Verschlüsselung

Die Amazon-Keyspaces-Verschlüsselung bietet erhöhte Sicherheit, indem alle Daten im Ruhezustand verschlüsselt werden, indem alle Daten im Ruhezustand mit Verschlüsselungsschlüsseln verschlüsselt werden, die in [AWS Key Management Service\(AWS KMS\)](#) gespeichert sind. Diese Funktionalität trägt zur Verringerung des Betriebsaufwands und der Komplexität bei, die mit dem Schutz sensibler Daten einhergeht. Wenn die Verschlüsselung im Ruhezustand ist, können Sie sicherheitssensible Anwendungen erstellen, die strenge Compliance-Anforderungen und behördliche Datenschutzerfordernisse erfüllen.

Die Amazon-Keyspaces-Verschlüsselung verschlüsselt Ihre Daten mit 256-bit-Advanced-Encryption-Standard (AES-256). Dies trägt dazu bei, Ihre Daten vor unautorisiertem Zugriff auf den zugrunde liegenden Speicher zu schützen.

Amazon Keyspaces verschlüsselt und entschlüsselt die Tabellendaten transparent. Amazon Keyspaces verwendet Umschlagverschlüsselung und eine Schlüsselhierarchie, um Datenverschlüsselungsschlüssel zu schützen. Es lässt sich in AWS KMS das Speichern und Verwalten des Root-Verschlüsselungsschlüssels integrieren. Weitere Informationen über die Verschlüsselungsschlüsselhierarchie finden Sie unter [the section called “Funktionsweise”](#). Weitere Informationen zu AWS KMS Konzepten wie der Envelope-Verschlüsselung finden Sie im AWS Key Management Service Developer Guide unter [AWS KMS Management Service-Konzepte](#).

Wenn Sie eine neue Tabelle erstellen, können Sie einen der folgenden AWS KMS Schlüssel (KMS-Schlüssel) auswählen:

- **AWS-eigener Schlüssel**— Dies ist der Standardverschlüsselungstyp. Der Schlüssel befindet sich im Besitz von Amazon Keyspaces (kein Aufpreis).
- **Kundenverwalteter Schlüssel** — Dieser Schlüssel wird in Ihrem Konto gespeichert und von Ihnen erstellt, besessen und verwaltet. Sie haben volle Kontrolle über den kundenverwalteten Schlüssel (AWS KMS-Gebühren werden erhoben).

Sie können jederzeit zwischen dem AWS-eigenen Schlüssel und dem kundenverwalteten Schlüssel wechseln. Sie können mit, oder Sie können mit, oder Sie können mit, oder Sie können mit, oder Sie können mit, oder Sie können mit CQL-Anweisungen einen kundenverwalteten Schlüssel angeben, wenn Sie eine neue Tabelle erstellen, oder Sie können den KMS-Schlüssel einer vorhandenen Tabelle wechseln. Um zu erfahren wie dies geht, vgl. [Verschlüsselung im Ruhezustand: So verwenden Sie vom Kunden verwaltete Schlüssel, um Tabellen in Amazon Keyspaces zu verschlüsseln](#).

**AWS-eigene Schlüssel** Die Verschlüsselung Für kundenverwaltete Schlüssel werden jedoch AWS KMS Gebühren erhoben. Weitere Informationen zu Preisen finden Sie unter [AWS KMS-Preise](#).

Die Amazon-Keyspace-Verschlüsselung ist in allen Regionen verfügbar AWS-Regionen, einschließlich AWS China (Peking) und AWS China (Ningxia). Weitere Informationen finden Sie unter [Verschlüsselung ruhender Daten: Funktionsweise in Amazon Keyspaces](#).

## Themen

- [Verschlüsselung ruhender Daten: Funktionsweise in Amazon Keyspaces](#)
- [Verschlüsselung im Ruhezustand: So verwenden Sie vom Kunden verwaltete Schlüssel, um Tabellen in Amazon Keyspaces zu verschlüsseln](#)

## Verschlüsselung ruhender Daten: Funktionsweise in Amazon Keyspaces

Die Verschlüsselung ruhender Daten durch Amazon Keyspaces (für Apache Cassandra) verschlüsselt Ihre Daten mithilfe des 256-Bit Advanced Encryption Standard (AES-256) verschlüsselt. Dies hilft, Ihre Daten vor unbefugtem Zugriff auf den zugrunde liegenden Speicher zu schützen. Alle Kundendaten in Amazon Keyspaces-Tabellen werden standardmäßig im Ruhezustand verschlüsselt, und die serverseitige Verschlüsselung ist transparent, sodass Änderungen an Anwendungen nicht erforderlich sind.

Die Verschlüsselung ruhender Daten ist für die Verwaltung des Verschlüsselungsschlüssels, mit dem Ihre Tabellen verschlüsselt werden, in AWS Key Management Service (AWS KMS) integriert. Wenn Sie eine neue Tabelle erstellen oder eine vorhandene Tabelle aktualisieren, können Sie eine der folgenden AWS KMS Schlüsseloptionen auswählen:

- **AWS-eigener Schlüssel**— Dies ist der Standardverschlüsselungstyp. Der Schlüssel befindet sich im Besitz von Amazon Keyspaces (kein Aufpreis)

- **Kundenverwalteter Schlüssel** — Dieser Schlüssel wird in Ihrem Konto gespeichert und von Ihnen erstellt, besessen und verwaltet. Sie haben volle Kontrolle über den kundenverwalteten Schlüssel (AWS KMS-Gebühren werden erhoben)

## AWS KMSSchlüssel (KMS-Schlüssel)

Die Verschlüsselung im Ruhezustand schützt alle Ihre Amazon Keyspaces-Daten mit einem AWS KMS Schlüssel. Standardmäßig verwendet Amazon Keyspaces einen [AWS-eigener Schlüssel](#), einen Mehrmandanten-Verschlüsselungsschlüssel, der in einem Amazon Keyspaces-Servicekonto erstellt und verwaltet wird.

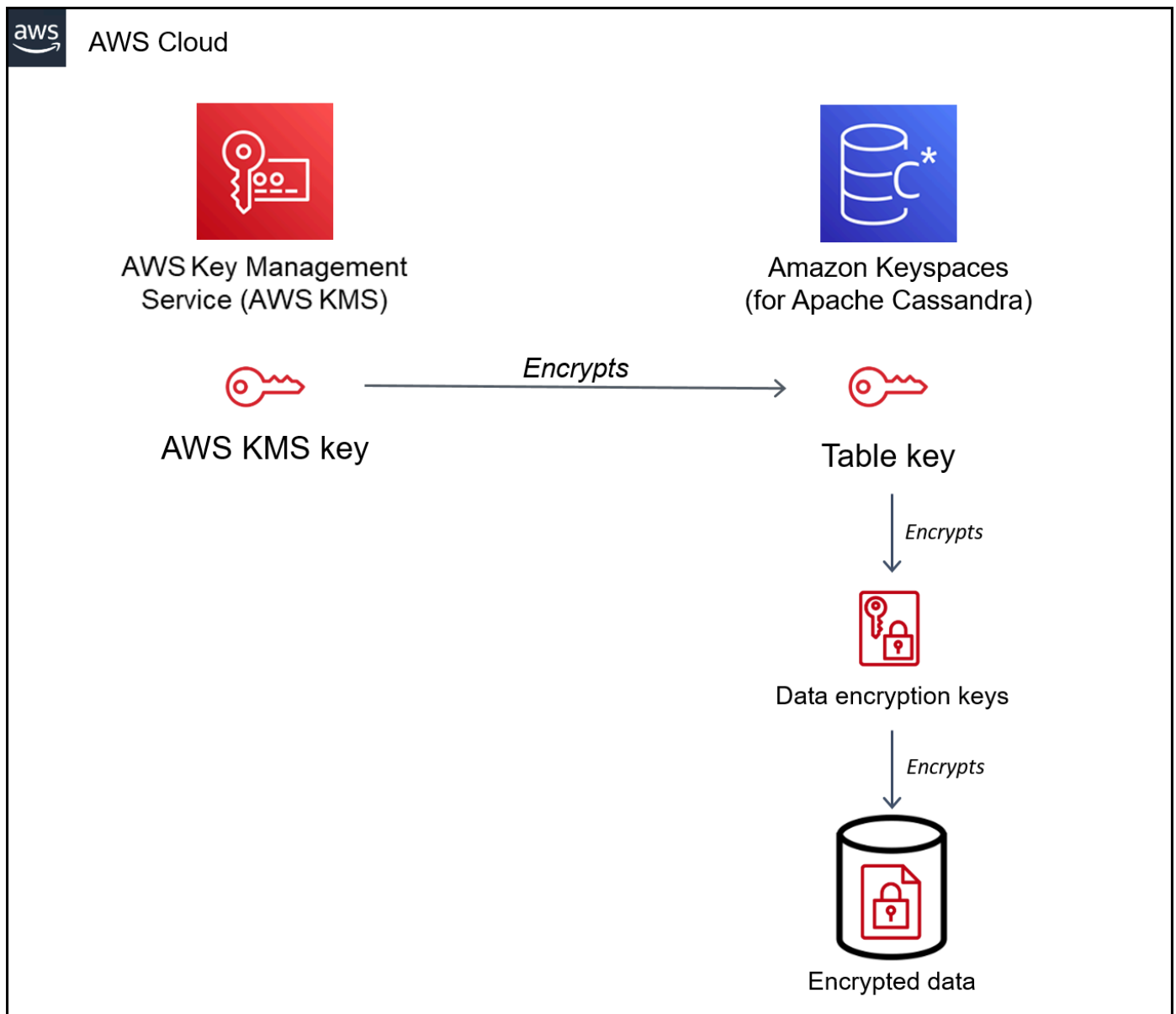
Sie können Ihre Amazon Keyspaces-Tabellen jedoch mit einem vom [Kunden verwalteten Schlüssel](#) in Ihrem verschlüsseln AWS-Konto. Sie können für jede Tabelle in einem Schlüsselraum einen anderen KMS-Schlüssel auswählen. Der KMS-Schlüssel, den Sie für eine Tabelle auswählen, wird auch zum Verschlüsseln aller Metadaten und wiederherstellbarer Sicherungen verwendet.

Sie wählen den KMS-Schlüssel für eine Tabelle aus, wenn Sie die Tabelle erstellen oder aktualisieren. Sie können den KMS-Schlüssel für eine Tabelle jederzeit entweder in der Amazon Keyspaces-Konsole oder mithilfe der [ALTER TABLE-Anweisung](#) ändern. Der Wechsel von KMS-Schlüsseln verläuft reibungslos und erfordert keine Ausfallzeiten und verursacht keine Serviceeinbußen.

## Hierarchie der Schlüssel

Amazon Keyspaces verwendet eine Schlüsselhierarchie zum Verschlüsseln von Daten. In dieser Schlüsselhierarchie ist der KMS-Schlüssel der Stammschlüssel. Es wird verwendet, um den Verschlüsselungsschlüssel der Amazon Keyspaces-Tabelle zu verschlüsseln und zu entschlüsseln. Der Tabellenverschlüsselungsschlüssel wird verwendet, um die Verschlüsselungsschlüssel zu verschlüsseln, die intern von Amazon Keyspaces zum Verschlüsseln und Entschlüsseln von Daten bei Lese- und Schreibvorgängen verwendet werden.

Mit der Verschlüsselungsschlüsselhierarchie können Sie Änderungen am KMS-Schlüssel vornehmen, ohne Daten erneut verschlüsseln zu müssen oder Anwendungen und den laufenden Datenbetrieb zu beeinträchtigen.



## Tabellen-Schlüssel

Der Tabellenschlüssel von Amazon Keyspaces wird als Schlüssel-Verschlüsselungsschlüssel verwendet. Amazon Keyspaces verwendet den Tabellenschlüssel zum Schutz der internen Datenverschlüsselungsschlüssel, die zum Verschlüsseln der in Tabellen, Protokolldateien und wiederherstellbaren Sicherungen gespeicherten Daten verwendet werden. Amazon Keyspaces generiert einen eindeutigen Datenverschlüsselungsschlüssel für jede zugrunde liegende Struktur in einer Tabelle generiert. Allerdings können mehrere Tabellenzeilen durch denselben Datenverschlüsselungsschlüssel geschützt werden.



Wenn Sie den KMS-Schlüssel zum ersten Mal auf einen vom Kunden verwalteten Schlüssel festlegen, AWS KMS wird ein Datenschlüssel generiert. Der AWS KMS Datenschlüssel bezieht sich auf den Tabellenschlüssel in Amazon Keyspaces.

Wenn Sie auf eine verschlüsselte Tabelle zugreifen, sendet Amazon Keyspaces eine Anforderung an AWS KMS um den KMS-Schlüssel zum Entschlüsseln des Tabellenschlüssels verwenden zu können. Anschließend verwendet es den Klartext-Tabellenschlüssel, um die Amazon Keyspaces-Datenverschlüsselungsschlüssel zu entschlüsseln, und es verwendet die Klartext-Datenverschlüsselungsschlüssel, um Tabellendaten zu entschlüsseln.

Amazon Keyspaces verwendet und speichert den Tabellenschlüssel und die Datenverschlüsselungsschlüssel außerhalb von AWS KMS. Alle Schlüssel werden mit [Advanced Encryption Standard](#) (AES)-Verschlüsselung und 256-Bit-Verschlüsselungsschlüsseln geschützt. Anschließend werden die verschlüsselten Schlüssel zusammen mit den verschlüsselten Daten gespeichert, sodass sie zur Entschlüsselung der Tabellendaten bei Bedarf zur Verfügung stehen.

### Tabellenschlüssel-Caching

Damit nicht AWS KMS für jede Amazon Keyspaces-Operation aufgerufen wird, wird die Klartext-Tabellenschlüssel für jede Verbindung zwischengespeichert. Wenn Amazon Keyspaces nach fünf Minuten Inaktivität eine Anforderung für den zwischengespeicherte Tabellenschlüssel erhält, sendet es eine neue Anforderung AWS KMS zum Entschlüsseln des Tabellenschlüssels an. Dieser Aufruf erfasst alle Änderungen, die in AWS KMS oder AWS Identity and Access Management (IAM) seit der letzten Anforderung zum Entschlüsseln des Tabellenschlüssels an den Zugriffsrichtlinien des KMS-Schlüssels vorgenommen wurden.

### Envelope-Verschlüsselung

Wenn Sie den kundenverwalteten Schlüssel für Ihre Tabelle ändern, generiert Amazon Keyspaces einen neuen Tabellenschlüssel. Anschließend verwendet es den neuen Tabellenschlüssel, um die Datenverschlüsselungsschlüssel erneut zu verschlüsseln. Es verwendet auch den neuen Tabellenschlüssel, um vorherige Tabellenschlüssel zu verschlüsseln, die zum Schutz wiederherstellbarer Backups verwendet werden. Dieser Prozess ist auch als Envelope-Verschlüsselung bekannt. Dadurch wird sichergestellt, dass Sie auf wiederherstellbare Backups zugreifen können, auch wenn Sie den vom Kunden verwalteten Schlüssel rotieren. Weitere Informationen zur Envelope-Verschlüsselung finden Sie unter [Envelope-Verschlüsselung](#) im AWS Key Management Service-Entwicklerhandbuch.

## Themen

- [AWSeigene Schlüssel](#)
- [Kundenverwaltete Schlüssel](#)
- [Nutzungshinweise zur Verschlüsselung ruhender Daten](#)

## AWSeigene Schlüssel

AWS-eigene Schlüssel sind nicht in deinem gespeichert AWS-Konto. Sie sind Teil einer Sammlung von KMS-Schlüsseln, die AWS besitzt und für die Verwendung in mehreren verwaltet AWS-Konten. AWS-Dienste, die Sie AWS-eigene Schlüssel zum Schutz Ihrer Daten verwenden können.

Sie können nicht anzeigen, verwalten oder verwenden AWS-eigene Schlüssel und auch nicht ihre Nutzung prüfen. Sie müssen jedoch keine Arbeit ausführen oder keine Programme ändern, um die Schlüssel zu schützen, die Ihre Daten verschlüsseln.

Es fällt keine monatliche Gebühr und auch keine Nutzungsgebühr für die Nutzung von AWS-eigene Schlüssel an. Sie werden AWS KMS auch nicht auf die -Limits für Ihr Konto angerechnet.

## Kundenverwaltete Schlüssel

Kundenverwaltete Schlüssel sind Schlüssel in Ihrem AWS-Konto die Sie erstellen, besitzen und verwalten. Sie haben die volle Kontrolle über diese KMS-Schlüssel.

Verwenden Sie einen kundenverwalteten KMS-Schlüssel, um die folgenden Funktionen zu erhalten:

- Sie erstellen und verwalten den vom Kunden verwalteten Schlüssel, einschließlich der Festlegung und Verwaltung der [wichtigsten Richtlinien](#), [IAM-Richtlinien](#) und Genehmigungen [zur Zugriffskontrolle auf den vom Kunden verwalteten Schlüssel](#). [Sie können den kundenverwalteten Schlüssel aktivieren und deaktivieren, die automatische Schlüsseldrehung aktivieren und deaktivieren und den kundenverwalteten Schlüssel zum Löschen planen, wenn er nicht mehr verwendet wird](#). Sie können Tags und Aliase für die von Ihnen verwalteten, vom Kunden verwalteten Schlüssel erstellen.
- Sie können einen kundenverwalteten Schlüssel mit [importiertem Schlüsselmaterial](#) oder einen kundenverwalteten Schlüssel in einem [benutzerdefinierten Schlüsselspeicher](#) verwenden, den Sie besitzen und verwalten.
- Sie können AWS CloudTrail und Amazon CloudWatch Logs verwenden, um die Anforderungen zu verfolgen, die Amazon Keyspaces in AWS KMS Ihrem Namen sendet. Weitere Informationen finden Sie unter [the section called "Schritt 6: Konfigurieren Sie die Überwachung mit AWS CloudTrail"](#).

Kundenverwaltete Schlüssel [fallen](#) für jeden API-Aufruf an, und für diese KMS-Schlüssel gelten AWS KMS -Kontingente. Weitere Informationen finden Sie unter [AWS KMS Ressourcen- oder Anforderungskontingente](#)

Wenn Sie einen kundenverwalteten Schlüssel als Stammverschlüsselungsschlüssel für eine Tabelle angeben, werden wiederherstellbare Sicherungen mit demselben Verschlüsselungsschlüssel verschlüsselt, der zum Zeitpunkt der Erstellung der Sicherung für die Tabelle angegeben wird. Wenn der KMS-Schlüssel für die Tabelle rotiert wird, stellt Key Enveloping sicher, dass der neueste KMS-Schlüssel Zugriff auf alle wiederherstellbaren Backups hat.

Amazon Keyspaces muss Zugriff auf Ihren vom Kunden verwalteten Schlüssel haben, um Ihnen Zugriff auf Ihre Tabellendaten zu gewähren. Wenn der Status des Verschlüsselungsschlüssels auf Deaktiviert gesetzt ist oder das Löschen geplant ist, kann Amazon Keyspaces keine Daten verschlüsseln oder entschlüsseln. Daher können Sie keine Lese- und Schreiboperationen für die Tabelle ausführen. Sobald der Dienst erkennt, dass der Zugriff auf den Verschlüsselungsschlüssel nicht möglich ist, sendet Amazon Keyspaces eine E-Mail-Benachrichtigung, um Sie zu benachrichtigen.

Sie müssen den Zugriff auf Ihren Verschlüsselungsschlüssel innerhalb von sieben Tagen wiederherstellen, andernfalls löscht Amazon Keyspaces Ihre Tabelle automatisch. Als Vorsichtsmaßnahme erstellt Amazon Keyspaces vor dem Löschen der Tabelle eine wiederherstellbare Sicherungskopie Ihrer Tabellendaten. Amazon Keyspaces bewahrt das wiederherstellbare Backup 35 Tage lang auf. Nach 35 Tagen können Sie Ihre Tabellendaten nicht mehr wiederherstellen. Das wiederherstellbare Backup wird Ihnen nicht in Rechnung gestellt, es fallen jedoch die üblichen [Wiederherstellungsgebühren an](#).

Sie können diese wiederherstellbare Sicherung verwenden, um Ihre Daten in einer neuen Tabelle wiederherzustellen. Um die Wiederherstellung zu initiieren, muss der letzte vom Kunden verwaltete Schlüssel aktiviert sein, der für die Tabelle verwendet wird, und Amazon Keyspaces muss Zugriff darauf haben.

#### Note

Wenn Sie eine Tabelle erstellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, auf den nicht zugegriffen werden kann oder der gelöscht werden soll, bevor der Erstellungsprozess abgeschlossen ist, tritt ein Fehler auf. Der Vorgang zum Erstellen der Tabelle schlägt fehl und Sie erhalten eine E-Mail-Benachrichtigung.

## Nutzungshinweise zur Verschlüsselung ruhender Daten

Beachten Sie Folgendes, wenn Sie die Verschlüsselung ruhender Daten in Amazon Keyspaces verwenden.

- Die serverseitige Verschlüsselung ruhender Daten ist für alle Amazon Keyspaces-Tabellen aktiviert und kann nicht deaktiviert werden. Die gesamte Tabelle ist im Ruhezustand verschlüsselt. Sie können keine bestimmten Spalten oder Zeilen für die Verschlüsselung auswählen.
- Standardmäßig verwendet Amazon Keyspaces einen einzigen Service-Standardschlüssel (AWS-eigener Schlüssel) für die Verschlüsselung all Ihrer Tabellen. Wenn dieser Schlüssel nicht existiert, wird er für Sie erstellt. Die Standardschlüssel des Dienstes können nicht deaktiviert werden.
- Bei der Verschlüsselung im Ruhezustand werden nur Daten verschlüsselt, solange sie statisch (im Ruhezustand) auf einem persistenten Speichermedium sind. Wenn die Datensicherheit bei der Übertragung oder der verwendeten Daten ein Problem darstellt, müssen Sie zusätzliche Maßnahmen ergreifen:
  - Daten während der Übertragung: Alle Ihre Daten in Amazon Keyspaces werden während der Übertragung verschlüsselt. Standardmäßig wird die Kommunikation zu und von Amazon Keyspaces durch Secure-Sockets-Layer- (SSL) -/Transport-Layer-Security- (TLS) - Verschlüsselung geschützt.
  - Genutzte Daten: Schützen Sie Daten durch Client-seitige Verschlüsselung, bevor sie an Amazon Keyspaces gesendet werden.
  - Kundenverwaltete Schlüssel: Die in Ihren Tabellen gespeicherten Daten werden immer mit Ihren vom Kunden verwalteten Schlüsseln verschlüsselt. Operationen, die atomare Aktualisierungen mehrerer Zeilen durchführen, verschlüsseln jedoch vorübergehend Daten, die AWS-eigene Schlüssel während der Verarbeitung verwendet werden. Dazu gehören Bereichslöschvorgänge und Operationen, die gleichzeitig auf statische und nicht statische Daten zugreifen.
- Für einen einzelnen, vom Kunden verwalteten Schlüssel können bis zu 50.000 [Zuschüsse gewährt](#) werden. Jede Amazon Keyspaces-Tabelle, die einem vom Kunden verwalteten Schlüssel zugeordnet ist, verbraucht 2 Zuschüsse. Ein Zuschuss wird freigegeben, wenn die Tabelle gelöscht wird. Der zweite Zuschuss wird verwendet, um einen automatischen Snapshot der Tabelle zu erstellen, um vor Datenverlust zu schützen, falls Amazon Keyspaces unbeabsichtigt den Zugriff auf den vom Kunden verwalteten Schlüssel verloren hat. Dieser Zuschuss wird 42 Tage nach dem Löschen der Tabelle freigegeben.

## Verschlüsselung im Ruhezustand: So verwenden Sie vom Kunden verwaltete Schlüssel, um Tabellen in Amazon Keyspaces zu verschlüsseln

Sie können die Konsolen- oder CQL-Anweisungen verwenden, um die AWS KMS key für neue Tabellen anzugeben und die Verschlüsselungsschlüssel vorhandener Tabellen in Amazon Keyspaces zu aktualisieren. Im folgenden Thema wird beschrieben, wie vom Kunden verwaltete Schlüssel für neue und bestehende Tabellen implementiert werden.

### Themen

- [Voraussetzungen: Erstellen Sie einen vom Kunden verwalteten Schlüssel mithilfe von Amazon Keyspaces AWS KMS und gewähren Sie diesen Berechtigungen](#)
- [Schritt 3: Geben eines vom Kunden verwalteten Schlüssel für eine neue Tabelle an](#)
- [Schritt 4: Den Verschlüsselungsschlüssel einer vorhandenen Tabelle aktualisieren](#)
- [Schritt 5: Verwenden Sie den Amazon Keyspaces-Verschlüsselungskontext in Protokollen](#)
- [Schritt 6: Konfigurieren Sie die Überwachung mit AWS CloudTrail](#)

**Voraussetzungen:** Erstellen Sie einen vom Kunden verwalteten Schlüssel mithilfe von Amazon Keyspaces AWS KMS und gewähren Sie diesen Berechtigungen

Bevor Sie eine Amazon Keyspaces-Tabelle mit einem vom [Kunden verwalteten Schlüssel](#) schützen können, müssen Sie zuerst den Schlüssel in AWS Key Management Service (AWS KMS) erstellen und dann Amazon Keyspaces autorisieren, diesen Schlüssel zu verwenden.

### Schritt 1: Erstellen eines kundenverwalteten Schlüssel mit AWS KMS

Um einen vom Kunden verwalteten Schlüssel zu erstellen, der zum Schutz einer Amazon Keyspaces-Tabelle verwendet werden soll, folgen Sie den Schritten unter [Erstellen symmetrischer Verschlüsselungs-KMS-Schlüssel](#) mithilfe der Konsole oder der AWS API.

### Schritt 2: Autorisieren Sie die Verwendung Ihres vom Kunden verwalteten Schlüssels

Bevor Sie einen [kundenverwalteten Schlüssel](#) zum Schutz einer Amazon Keyspaces-Tabelle auswählen können, müssen die Richtlinien für diesen kundenverwalteten Schlüssel Amazon Keyspaces zu seiner Verwendung in Ihrem Namen berechtigen. Sie haben volle Kontrolle über die Richtlinien und Erteilungen für den vom Kunden verwalteten Schlüssel. Sie können diese Berechtigungen in einer [Schlüsselrichtlinie](#), einer [IAM-Richtlinie](#) oder einer [Ertelung](#) bereitstellen.

Amazon Keyspaces benötigt keine zusätzliche Autorisierung, um die Amazon Keyspaces-Tabellen in Ihrem AWS Konto mit der Standardeinstellung [AWS-eigener Schlüssel](#) zu schützen.

In den folgenden Themen wird gezeigt, wie Sie die erforderlichen Berechtigungen mithilfe von IAM-Richtlinien und Zuschüssen konfigurieren, die es Amazon Keyspaces-Tabellen ermöglichen, einen vom Kunden verwalteten Schlüssel zu verwenden.

## Themen

- [Schlüsselverwaltete](#)
- [Beispiel für eine Schlüsselrichtlinie](#)
- [Verwenden von Erteilungen zum Autorisieren von Amazon Keyspaces](#)

## Schlüsselverwaltete

Wenn Sie einen [kundenverwalteten Schlüssel](#) zum Schutz einer Amazon Keyspaces-Tabelle auswählen, erhält Amazon Keyspaces die Berechtigung, den kundenverwalteten Schlüssel im Namen des Prinzipals zu verwenden, der die Auswahl trifft. Dieser Prinzipal, ein Benutzer oder eine Rolle, muss über die Berechtigungen für den kundenverwalteten Schlüssel verfügen, die Amazon Keyspaces benötigt.

Amazon Keyspaces erfordert mindestens die folgenden Berechtigungen für einen kundenverwalteten Schlüssel:

- [kms:Encrypt](#)
- [kms:Decrypt](#)
- [kms:ReEncrypt](#) \* (für [kms:ReEncryptFrom](#) und [kms:ReEncryptTo](#))
- [kms:GenerateDataKey](#) \* (für [kms:GenerateDataKey](#) und [kms:GenerateDataKeyWithoutPlaintext](#))
- [km:DescribeKey](#)
- [km:CreateGrant](#)

## Beispiel für eine Schlüsselrichtlinie

Beispielsweise bietet die folgende Beispiel-Schlüsselrichtlinie nur die erforderlichen Berechtigungen. Die Richtlinie hat folgende Auswirkungen:

- Erlaubt Amazon Keyspaces die Verwendung des vom Kunden verwalteten Schlüssel für kryptografische Operationen und die Erstellung von Erteilungen — jedoch nur, wenn es im Auftrag

von Prinzipalen im Konto handelt, die über die Berechtigung zur Verwendung von Amazon Keyspaces verfügen. Wenn die in der Richtlinienanweisung angegebenen Prinzipale nicht zur Verwendung von Amazon Keyspaces berechtigt sind, schlägt der Anruf selbst dann fehl, wenn er vom Amazon Keyspaces-Service stammt.

- Der [Bedingungsschlüssel kms:ViaService](#) erlaubt die Berechtigungen nur, wenn die Anforderung von Amazon Keyspaces im Auftrag der in der Richtlinienanweisung aufgeführten Prinzipale stammt. Diese Prinzipale können diese Operationen nicht direkt aufrufen. Beachten Sie, dass der kms:ViaService-Wert, `cassandra.*.amazonaws.com`, in der Region-Position ein Sternchen (\*) hat. Amazon Keyspaces benötigt die Erlaubnis, unabhängig von einer bestimmten Person zu sein AWS-Region.
- Gewährt den kundenverwalteten Schlüsselverwalteten (Benutzer, die die `diedb-team` Rolle annehmen können) schreibgeschützten Zugriff auf den kundenverwalteten Schlüssel und die Berechtigung, Erteilungen zu widerrufen, einschließlich der Erteilungen, [die Amazon Keyspaces zum Schutz der Tabelle benötigt](#).
- Gewährt Amazon Keyspaces schreibgeschützten Zugriff auf den kundenverwalteten Schlüssel. In diesem Fall kann Amazon Keyspaces diese Operationen direkt aufrufen. Es muss nicht im Namen eines Kontoinhabers handeln.

Bevor Sie eine Beispiel-Schlüsselrichtlinie verwenden, ersetzen Sie die Beispiel-Prinzipale durch tatsächliche Prinzipale aus Ihrem AWS-Konto.

```
{
 "Id": "key-policy-cassandra",
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "Allow access through Amazon Keyspaces for all principals in the account that are authorized to use Amazon Keyspaces",
 "Effect": "Allow",
 "Principal": {"AWS": "arn:aws:iam::111122223333:user/db-lead"},
 "Action": [
 "kms:Encrypt",
 "kms:Decrypt",
 "kms:ReEncrypt*",
 "kms:GenerateDataKey*",
 "kms:DescribeKey",
 "kms:CreateGrant"
],
 "Resource": "*"
 }
]
}
```

```

 "Condition": {
 "StringLike": {
 "kms:ViaService" : "cassandra.*.amazonaws.com"
 }
 },
 {
 "Sid": "Allow administrators to view the customer managed key and revoke grants",
 "Effect": "Allow",
 "Principal": {
 "AWS": "arn:aws:iam::111122223333:role/db-team"
 },
 "Action": [
 "kms:Describe*",
 "kms:Get*",
 "kms:List*",
 "kms:RevokeGrant"
],
 "Resource": "*"
 }
]
}

```

## Verwenden von Erteilungen zum Autorisieren von Amazon Keyspaces

Zusätzlich zu den wichtigsten Richtlinien verwendet Amazon Keyspaces Erteilungen, um Berechtigungen für einen kundenverwalteten Schlüssel festzulegen. Verwenden Sie den [ListGrants](#) Vorgang, um die Zuschüsse für einen vom Kunden verwalteten Schlüssel in Ihrem Konto einzusehen. Amazon Keyspaces benötigt keine Erteilungen oder zusätzliche Berechtigungen, um die zum Schutz Ihrer Tabelle [AWS-eigener Schlüssel](#) zu verwenden.

Amazon Keyspaces verwendet die Berechtigungen zum Ausführen von Hintergrundsystemwartung und kontinuierlichen Datenschutzaufgaben. Außerdem werden Erteilungen verwendet, um Tabellenschlüssel zu generieren.

Jede Erteilung ist tabellenspezifisch. Wenn das Konto mehrere Tabellen enthält, die mit demselben vom Kunden verwalteten Schlüssel verschlüsselt sind, wird für jede Tabelle eine Erteilung für jeden Typ gewährt. Die Gewährung ist durch den [Amazon Keyspaces-Verschlüsselungskontext](#) eingeschränkt, der den Tabellennamen und die AWS-Konto ID umfasst. Der Zuschuss beinhaltet die Erlaubnis, [den Zuschuss zurückzuziehen](#), wenn er nicht mehr benötigt wird.



Zum Erstellen von Erteilungen muss Amazon Keyspaces über die Berechtigung verfügen, `CreateGrant` im Namen des Benutzers anzurufen, der die verschlüsselte Tabelle erstellt hat.

Die Schlüsselrichtlinie kann es dem Konto auch erlauben, die [Erteilung für den vom Kunden verwalteten Schlüssel zu widerrufen](#). Wenn Sie jedoch die Erteilung für eine aktive verschlüsselte Tabelle widerrufen, kann Amazon Keyspaces die Tabelle nicht mehr schützen und pflegen.

Schritt 3: Geben eines vom Kunden verwalteten Schlüssel für eine neue Tabelle an

Führen Sie die folgenden Schritte aus, um den kundenverwalteten Schlüssel in einer neuen Tabelle mithilfe der Amazon Keyspaces-Konsole oder CQL anzugeben.

Erstellen einer verschlüsselten Tabelle mit einem vom Kunden verwalteten Schlüssel (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich Tables (Tabellen) und anschließend Create table (Tabelle erstellen) aus.
3. Wählen Sie auf der Seite Tabelle erstellen im Abschnitt Tabellendetails ein Schlüsselfeld aus und geben Sie einen Namen für die neue Tabelle ein.
4. Erstellen Sie im Abschnitt Schema das Schema für Ihre Tabelle.
5. Wählen Sie im Abschnitt Tabelleneinstellungen die Option Einstellungen anpassen aus.
6. Fahren Sie mit den Verschlüsselungseinstellungen fort.

In diesem Schritt wählen Sie die Verschlüsselungseinstellungen für die Tabelle aus.

Wählen Sie im Abschnitt Verschlüsselung im Ruhezustand unter Wählen Sie einen AWS KMS key anderen KMS-Schlüssel (erweitert) und wählen Sie im Suchfeld einen Amazon-Ressourcennamen (ARN) aus AWS KMS key oder geben Sie einen Amazon-Ressourcennamen (ARN) ein.

#### Note

Wenn auf den von Ihnen ausgewählten Schlüssel nicht zugegriffen werden kann oder die erforderlichen Berechtigungen fehlen, finden Sie im AWS Key Management Service Entwicklerhandbuch unter [Problembehandlung beim Schlüsselzugriff](#) weitere Informationen.

## 7. Wählen Sie Create (Erstellen) , um die verschlüsselte Tabelle zu erstellen.

Erstellen einer neuen Tabelle mit einem kundenverwalteten Schlüssel für Verschlüsselung im Ruhezustand (CQL)

Um eine neue Tabelle zu erstellen, die einen kundenverwalteten Schlüssel für die Verschlüsselung im Ruhezustand verwendet, können Sie die `CREATE TABLE` Anweisung wie im folgenden Beispiel gezeigt verwenden. Stellen Sie sicher, dass Sie den Schlüssel-ARN durch einen ARN ersetzen, um einen gültigen Schlüssel mit Berechtigungen zu erhalten, die Amazon Keyspaces gewährt wurden.

```
CREATE TABLE my_keyspace.my_table(id bigint, name text, place text STATIC, PRIMARY
KEY(id, name)) WITH CUSTOM_PROPERTIES = {
 'encryption_specification':{
 'encryption_type': 'CUSTOMER_MANAGED_KMS_KEY',
 'kms_key_identifier': 'arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
};
```

Wenn Sie eine erhalten `Invalid Request Exception`, müssen Sie bestätigen, dass der vom Kunden verwaltete Schlüssel gültig ist und Amazon Keyspaces über die erforderlichen Berechtigungen verfügt. Um zu überprüfen, ob der Schlüssel korrekt konfiguriert wurde, finden Sie im [AWS Key Management Service Entwicklerhandbuch](#) unter [Problembehandlung beim Schlüsselzugriff](#).

### Schritt 4: Den Verschlüsselungsschlüssel einer vorhandenen Tabelle aktualisieren

Sie können auch die Amazon Keyspaces-Konsole oder CQL verwenden, um die Verschlüsselungsschlüssel einer vorhandenen Tabelle jederzeit zwischen einem vom Kunden verwalteten KMS-Schlüssel AWS-eigener Schlüssel und einem vom Kunden verwalteten KMS-Schlüssel zu ändern.

Aktualisieren Sie eine vorhandene Tabelle mit dem neuen vom Kunden verwalteten Schlüssel (Konsole)

1. Melden Sie sich bei der [AWS Management Console](#) an und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im Navigationsbereich **Tables (Tabellen)** aus.

3. Wählen Sie die Tabelle aus, die Sie aktualisieren möchten, und wählen Sie dann die Registerkarte Zusätzliche Einstellungen.
4. Wählen Sie im Abschnitt Verschlüsselung im Ruhezustand die Option Verschlüsselung verwalten aus, um die Verschlüsselungseinstellungen für die Tabelle zu bearbeiten.

Wählen Sie unter AWS KMS key Choose an die Option Einen anderen KMS-Schlüssel auswählen (erweitert) und wählen Sie im Suchfeld einen Amazon-Ressourcennamen (ARN) aus AWS KMS key oder geben Sie einen Amazon-Ressourcennamen (ARN) ein.

#### Note

Wenn der von Ihnen gewählte Schlüssel nicht gültig ist, lesen Sie den Abschnitt [Problembehandlung beim Schlüsselzugriff](#) im AWS Key Management Service Entwicklerhandbuch.

Alternativ können Sie einen AWS-eigener Schlüssel für eine Tabelle auswählen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist.

5. Wählen Sie Änderungen speichern, um Ihre Änderungen an der Tabelle zu speichern.

Aktualisieren des Verschlüsselungsschlüssels, der für eine vorhandene Tabelle verwendet wurde

Um den Verschlüsselungsschlüssel einer vorhandenen Tabelle zu ändern, verwenden Sie die ALTER TABLE Anweisung, um einen vom Kunden verwalteten Schlüssel für die Verschlüsselung im Ruhezustand anzugeben. Stellen Sie sicher, dass Sie den Schlüssel-ARN durch einen ARN ersetzen, um einen gültigen Schlüssel mit Berechtigungen zu erhalten, die Amazon Keyspaces gewährt wurden.

```
ALTER TABLE my_keyspace.my_table WITH CUSTOM_PROPERTIES = {
 'encryption_specification':{
 'encryption_type': 'CUSTOMER_MANAGED_KMS_KEY',
 'kms_key_idenfifier': 'arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
};
```

Wenn Sie eine erhalten `Invalid Request Exception`, müssen Sie bestätigen, dass der vom Kunden verwaltete Schlüssel gültig ist und Amazon Keyspaces über die erforderlichen

Berechtigungen verfügt. Um zu überprüfen, ob der Schlüssel korrekt konfiguriert wurde, finden Sie im AWS Key Management Service Entwicklerhandbuch unter [Problembehandlung beim Schlüsselzugriff](#).

Um den Verschlüsselungsschlüssel wieder auf die Standardoption Encryption at Rest mit zu ändern AWS-eigene Schlüssel, können Sie die ALTER TABLE Anweisung verwenden, wie im folgenden Beispiel gezeigt.

```
ALTER TABLE my_keyspace.my_table WITH CUSTOM_PROPERTIES = {
 'encryption_specification': {
 'encryption_type' : 'AWS_OWNED_KMS_KEY'
 }
};
```

Schritt 5: Verwenden Sie den Amazon Keyspaces-Verschlüsselungskontext in Protokollen

Ein [Verschlüsselungskontext](#) ist eine Gruppe von Schlüssel/Wert-Paaren mit willkürlichen, nicht geheimen Daten. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, bindet AWS KMS den Verschlüsselungskontext kryptografisch an die verschlüsselten Daten. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

Amazon Keyspaces verwendet in allen AWS KMS kryptografischen Operationen denselben Verschlüsselungskontext. Wenn Sie einen [kundenverwalteten Schlüssel](#) zum Schutz Ihrer Amazon Keyspaces-Tabelle verwenden, können Sie anhand des Verschlüsselungskontexts die Verwendung des vom Kunden verwalteten Schlüssel in Prüfungs-Datensätzen und -Protokollen identifizieren. Es erscheint auch im Klartext in Protokollen, z. B. in Protokollen für [AWS CloudTrail](#) und [Amazon CloudWatch Logs](#).

In den Anforderungen an AWS KMS verwendet Amazon Keyspaces einen Verschlüsselungskontext mit drei Schlüssel-Wert-Paaren.

```
"encryptionContextSubset": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "mytable"
 "aws:cassandra:subscriberId": "111122223333"
}
```

- **Keyspace** — Das erste Schlüssel-Wert-Paar gibt den Schlüsselraum an, der die Tabelle enthält, die Amazon Keyspaces verschlüsselt. Der Schlüssel lautet `aws:cassandra:keyspaceName`. Der Wert ist der Name des Keyspace.

```
"aws:cassandra:keyspaceName": "<keyspace-name>"
```

Beispiel:

```
"aws:cassandra:keyspaceName": "my_keyspace"
```

- **Tabelle**: Das zweite Schlüssel-Wert-Paar gibt die Tabelle an, die Amazon Keyspaces verschlüsselt. Der Schlüssel lautet `aws:cassandra:tableName`. Der Wert ist der Name der Tabelle.

```
"aws:cassandra:tableName": "<table-name>"
```

Beispiel:

```
"aws:cassandra:tableName": "my_table"
```

- **Konto** — Das dritte Schlüssel-Wert-Paar gibt das anAWS-Konto. Der Schlüssel lautet `aws:cassandra:subscriberId`. Der Wert ist die Konto-ID.

```
"aws:cassandra:subscriberId": "<account-id>"
```

Beispiel:

```
"aws:cassandra:subscriberId": "111122223333"
```

## Schritt 6: Konfigurieren Sie die Überwachung mitAWS CloudTrail

Wenn Sie einen [kundenverwalteten Schlüssel](#) zum Schutz Ihrer Amazon-Keyspaces-Tabellen verwenden, können Sie mithilfe vonAWS CloudTrail -Protokollen, die Anforderungen nachverfolgen, die Amazon Keyspaces in Ihrem NamenAWS KMS an sendet.

DieGenerateDataKey,DescribeKeyDecrypt, undCreateGrant Anfragen werden in diesem Abschnitt behandelt. Darüber hinaus verwendet Amazon Keyspaces einen [RetireGrant](#)Vorgang, um einen Zuschuss zu entfernen, wenn Sie eine Tabelle löschen.

## GenerateDataKey

Amazon Keyspaces erstellt einen eindeutigen Tabellenschlüssel zum Verschlüsseln von Daten im Ruhezustand. Es sendet eine [GenerateDataKey](#)Anfrage an AWS KMS, die den KMS-Schlüssel für die Tabelle angibt.

Das Ereignis, das die GenerateDataKey-Operation aufzeichnet, ähnelt dem folgenden Beispiereignis. Der Benutzer ist das Amazon Keyspaces-Dienstkonto. Die Parameter enthalten den Amazon-Ressourcennamen (ARN) des vom Kunden verwalteten Schlüssel, einen Schlüsselbezeichner, der einen 256-Bit-Schlüssel benötigt, und den [Verschlüsselungskontext](#), der den Schlüsselverwalteten, die Tabelle und das identifiziert AWS-Konto.

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AWSService",
 "invokedBy": "AWS Internal"
 },
 "eventTime": "2021-04-16T04:56:05Z",
 "eventSource": "kms.amazonaws.com",
 "eventName": "GenerateDataKey",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "keySpec": "AES_256",
 "encryptionContext": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "my_table",
 "aws:cassandra:subscriberId": "123SAMPLE012"
 }
 },
 "keyId": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111"
},
"responseElements": null,
"requestID": "5e8e9cb5-9194-4334-aacc-9dd7d50fe246",
"eventID": "49fccab9-2448-4b97-a89d-7d5c39318d6f",
"readOnly": true,
"resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
```

```

 "ARN": "arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123SAMPLE012",
 "sharedEventID": "84fbaaf0-9641-4e32-9147-57d2cb08792e"
}

```

## DescribeKey

Amazon Keyspaces verwendet einen [DescribeKey](#) Vorgang, um zu ermitteln, ob der von Ihnen ausgewählte KMS-Schlüssel im Konto und in der Region vorhanden ist.

Das Ereignis, das die DescribeKey-Operation aufzeichnet, ähnelt dem folgenden Beispiereignis. Der Benutzer ist das Amazon Keyspaces-Dienstkonto. Die Parameter enthalten den ARN des vom Kunden verwalteten Schlüssel und einen Schlüsselbezeichner, der einen 256-Bit-Schlüssel benötigt.

```

{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AIDAZ3FNIIIVIZZ6H7CFQG",
 "arn": "arn:aws:iam::123SAMPLE012:user/admin",
 "accountId": "123SAMPLE012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "userName": "admin",
 "sessionContext": {
 "sessionIssuer": {},
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-04-16T04:55:42Z"
 }
 }
 },
 "invokedBy": "AWS Internal"
},
"eventTime": "2021-04-16T04:55:58Z",
"eventSource": "kms.amazonaws.com",
"eventName": "DescribeKey",

```

```

 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "keyId": "arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 },
 "responseElements": null,
 "requestID": "c25a8105-050b-4f52-8358-6e872fb03a6c",
 "eventID": "0d96420e-707e-41b9-9118-56585a669658",
 "readOnly": true,
 "resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123SAMPLE012"
 }

```

## Decrypt

Wenn Sie auf eine Amazon-Keyspaces-Tabelle zugreifen, muss Amazon Keyspaces den Tabellenschlüssel entschlüsseln, damit die in der Hierarchie tiefer angeordneten Schlüssel entschlüsselt werden können. Anschließend werden die Daten in der Tabelle entschlüsselt. Zum Entschlüsseln des Tabellenschlüssels sendet Amazon Keyspaces eine [Decrypt-Anforderung](#) an AWS KMS, die den KMS-Schlüssel für die Tabelle angibt.

Das Ereignis, das die Decrypt-Operation aufzeichnet, ähnelt dem folgenden Beispielergebnis. Der Benutzer ist der Prinzipal in Ihrem AWS-Konto, mit dem auf die Tabelle zugegriffen wird. Die Parameter enthalten den verschlüsselten Tabellenschlüssel (als Verschlüsselungstext-Blob) und den [Verschlüsselungskontext](#), der die Tabelle und das identifiziert AWS-Konto. AWS KMS leitet die ID des vom Kunden verwalteten Schlüssels aus dem Chiffretext ab.

```

{
 "eventVersion": "1.08",
 "userIdentity": {

```



```

 "type": "AWSService",
 "invokedBy": "AWS Internal"
 },
 "eventTime": "2021-04-16T05:29:44Z",
 "eventSource": "kms.amazonaws.com",
 "eventName": "Decrypt",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "encryptionContext": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "my_table",
 "aws:cassandra:subscriberId": "123SAMPLE012"
 },
 "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
 },
 "responseElements": null,
 "requestID": "50e80373-83c9-4034-8226-5439e1c9b259",
 "eventID": "8db9788f-04a5-4ae2-90c9-15c79c411b6b",
 "readOnly": true,
 "resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123SAMPLE012",
 "sharedEventID": "7ed99e2d-910a-4708-a4e3-0180d8dbb68e"
}

```

## CreateGrant

Wenn Sie einen [kundenverwalteten Schlüssel](#) zum Schutz Ihrer Amazon Keyspaces-Tabelle verwenden, verwendet Amazon Keyspaces [Erteilungen](#), um dem Service die Ausführung von Aufgaben für kontinuierlichen Datenschutz, Wartung und Haltbarkeit zu erlauben. Diese Zuschüsse sind nicht erforderlich für [AWS-eigene Schlüssel](#).

Die von Amazon Keyspaces erstellten Erteilungen sind tabellenspezifisch. Der Principal in der [CreateGrant](#)Anfrage ist der Benutzer, der die Tabelle erstellt hat.

Das Ereignis, das die CreateGrant-Operation aufzeichnet, ähnelt dem folgenden Beispielergebnis. Die Parameter enthalten den ARN des vom Kunden verwalteten Schlüssel für die Tabelle, den Empfänger-Prinzipal und den ausscheidenden Prinzipal (der Amazon Keyspaces-Service) sowie die Operationen, die der Erteilung abdeckt. Es enthält auch eine Einschränkung, nach der alle Verschlüsselungsvorgänge den angegebenen [Verschlüsselungskontext](#) verwenden müssen.

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AIDAZ3FNIIVIZZ6H7CFQG",
 "arn": "arn:aws:iam::arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111:user/admin",
 "accountId": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111",
 "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
 "userName": "admin",
 "sessionContext": {
 "sessionIssuer": {},
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-04-16T04:55:42Z"
 }
 }
 },
 "invokedBy": "AWS Internal"
},
"eventTime": "2021-04-16T05:11:10Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-east-1",
"sourceIPAddress": "AWS Internal",
"userAgent": "AWS Internal",
"requestParameters": {
 "keyId": "a7d328af-215e-4661-9a69-88c858909f20",
 "operations": [
 "DescribeKey",
 "GenerateDataKey",
 "Decrypt",
```

```

 "Encrypt",
 "ReEncryptFrom",
 "ReEncryptTo",
 "RetireGrant"
],
 "constraints": {
 "encryptionContextSubset": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "my_table",
 "aws:cassandra:subscriberId": "123SAMPLE012"
 }
 },
 "retiringPrincipal": "cassandratest.us-east-1.amazonaws.com",
 "granteePrincipal": "cassandratest.us-east-1.amazonaws.com"
},
"responseElements": {
 "grantId":
"18e4235f1b07f289762a31a1886cb5efd225f069280d4f76cd83b9b9b5501013"
},
"requestID": "b379a767-1f9b-48c3-b731-fb23e865e7f7",
"eventID": "29ee1fd4-28f2-416f-a419-551910d20291",
"readOnly": false,
"resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123SAMPLE012"
}

```

## Verschlüsselung während der Übertragung in Amazon Keyspaces

Amazon Keyspaces akzeptiert nur sichere Verbindungen mit Transport Layer Security (TLS). Die Verschlüsselung während der Übertragung bietet eine zusätzliche Datenschutzebene, indem Ihre Daten während der Übertragung von und zu Amazon Keyspaces verschlüsselt werden. Unternehmensrichtlinien, Branchen- oder Regierungsvorschriften sowie Compliance-Anforderungen

erfordern häufig die Verwendung von Verschlüsselung bei der Übertragung, um die Datensicherheit Ihrer Anwendungen bei der Übertragung von Daten über das Netzwerk zu erhöhen.

Informationen zum Verschlüsseln von `cqlsh` Verbindungen zu Amazon Keyspaces mithilfe von TLS finden Sie unter [the section called “Wie konfiguriert man `cqlsh` Verbindungen für TLS manuell”](#). Weitere Informationen zur Verwendung der TLS-Verschlüsselung mit Clienttreibern finden Sie unter [the section called “Verwenden eines Cassandra-Client-Treibers”](#).

## Datenschutz im Netzwerkverkehr in Amazon Keyspaces

In diesem Thema wird beschrieben, wie Amazon Keyspaces (für Apache Cassandra) Verbindungen von lokalen Anwendungen zu Amazon Keyspaces sowie zwischen Amazon Keyspaces und anderen Ressourcen innerhalb derselben sichert. AWS AWS-Region

### Datenverkehr zwischen Service und On-Premises-Clients und -Anwendungen

Sie haben zwei Verbindungsoptionen zwischen Ihrem privaten Netzwerk und AWS:

- AWS Site-to-Site VPN-Verbindung Weitere Informationen finden Sie unter [Was ist AWS Site-to-Site VPN?](#) im AWS Site-to-Site VPN-Benutzerhandbuch.
- AWS Direct Connect-Verbindung Weitere Informationen finden Sie unter [Was ist AWS Direct Connect?](#) im AWS Direct Connect-Benutzerhandbuch.

Als verwalteter Service ist Amazon Keyspaces (für Apache Cassandra) durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter [AWS Cloud-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.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Keyspaces zuzugreifen. 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.

Amazon Keyspaces unterstützt zwei Methoden zur Authentifizierung von Kundenanfragen. Die erste Methode verwendet dienstspezifische Anmeldeinformationen, bei denen es sich um kennwortbasierte Anmeldeinformationen handelt, die für einen bestimmten IAM-Benutzer generiert wurden. Sie können das Passwort mithilfe der IAM-Konsole, der oder der AWS API erstellen AWS CLI und verwalten. Weitere Informationen finden Sie unter [Verwenden von IAM mit Amazon Keyspaces](#).

Die zweite Methode verwendet ein Authentifizierungs-Plugin für den DataStax Open-Source-Java-Treiber für Cassandra. Dieses Plugin ermöglicht es [IAM-Benutzern, Rollen und Verbundidentitäten](#), mithilfe des [AWSSignature Version 4-Prozesses](#) (SIGv4) Authentifizierungsinformationen zu Amazon Keyspaces (für Apache Cassandra) API-Anfragen hinzuzufügen. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

## Datenverkehr zwischen AWS-Ressourcen in derselben Region

Schnittstellen-VPC-Endpunkte ermöglichen die private Kommunikation zwischen Ihrer Virtual Private Cloud (VPC), die in Amazon VPC und Amazon Keyspaces ausgeführt wird. Interface-VPC-Endpunkte werden von unterstützt. Dabei handelt es sich um einen AWS DienstAWS PrivateLink, der die private Kommunikation zwischen VPCs und Diensten ermöglicht. AWS AWS PrivateLinkermöglicht dies, indem Sie eine elastische Netzwerkschnittstelle mit privaten IPs in Ihrer VPC verwenden, sodass der Netzwerkverkehr das Amazon-Netzwerk nicht verlässt. Schnittstellen-VPC-Endpunkte benötigen keinen Internet-Gateway, kein NAT-Gerät, keine VPN-Verbindung und keine AWS Direct Connect-Verbindung. Weitere Informationen finden Sie unter [Amazon Virtual Private Cloud and Interface VPC Endpoints \(\) AWS PrivateLink](#). Beispiele für Richtlinien finden Sie unter [the section called “Verwenden von Schnittstellen-VPC-Endpunkten für Amazon Keyspaces”](#).

## AWS Identity and Access Management für Amazon Keyspaces

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon Keyspaces-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

### Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert Amazon Keyspaces mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien von Amazon Keyspaces](#)
- [AWSverwaltete Richtlinien für Amazon Keyspaces](#)
- [Fehlerbehebung bei Identität und Zugriff auf Amazon Keyspaces](#)
- [Verwenden von serviceverknüpften Rollen für Amazon Keyspaces](#)

## Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon Keyspaces ausführen.

**Servicebenutzer** — Wenn Sie den Amazon Keyspaces-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Funktionen von Amazon Keyspaces verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie in Amazon Keyspaces nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unter [Fehlerbehebung bei Identität und Zugriff auf Amazon Keyspaces](#).

**Service-Administrator** — Wenn Sie in Ihrem Unternehmen für die Amazon Keyspaces-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon Keyspaces. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon Keyspaces Ihre Servicebenutzer 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 darüber, wie Ihr Unternehmen IAM mit Amazon Keyspaces verwenden kann, finden Sie unter [So funktioniert Amazon Keyspaces mit IAM](#).

**IAM-Administrator** — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon Keyspaces schreiben können. Beispiele für identitätsbasierte Amazon Keyspaces-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien von Amazon Keyspaces](#).

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, 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 AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, 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.

## IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb von Ihrem AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. 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 innerhalb Ihres Unternehmens AWS-Konto, die über bestimmte Berechtigungen verfügt. 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 Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie unter [Kontenübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch](#).
- **Serviceübergreifender Zugriff** — Einige verwenden Funktionen in anderen. AWS-Services 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 auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. 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.

- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer Service-Verknüpfung verbunden ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-Verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon EC2 ausgeführte Anwendungen** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen. AWS CLI ist eher zu empfehlen, als ein Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS-Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt 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 kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden 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 mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff 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 von der AWS Management Console AWS CLI, der oder 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 System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören 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](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. 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, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete 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 Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte 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. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. 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. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

## So funktioniert Amazon Keyspaces mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon Keyspaces zu verwalten, sollten Sie wissen, welche IAM-Funktionen für Amazon Keyspaces verfügbar sind. Einen allgemeinen Überblick darüber, wie Amazon Keyspaces und andere AWS Dienste mit IAM funktionieren, finden Sie im [AWS IAM-Benutzerhandbuch unter Dienste, die mit IAM funktionieren](#).

### Themen

- [Identitätsbasierte Richtlinien von Amazon Keyspaces](#)
- [Ressourcenbasierte Richtlinien von Amazon Keyspaces](#)
- [Autorisierung basierend auf Amazon Keyspaces-Tags](#)
- [IAM-Rollen in Amazon Keyspaces](#)

## Identitätsbasierte Richtlinien von Amazon Keyspaces

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. Amazon Keyspaces unterstützt bestimmte Aktionen und Ressourcen sowie Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Die servicespezifischen Ressourcen und Aktionen von Amazon Keyspaces sowie Bedingungskontextschlüssel, die für IAM-Berechtigungsrichtlinien verwendet werden können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Keyspaces \(für Apache Cassandra\)](#) in der Service Authorization Reference.

## Aktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, 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 der zugehörige AWS API-Vorgang. 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.

Richtlinienaktionen in Amazon Keyspaces verwenden das folgende Präfix vor der Aktion: `cassandra:`. Um beispielsweise jemandem die Erlaubnis zu erteilen, einen Amazon Keyspaces-Schlüsselraum mit der Amazon CREATE Keyspaces-CQL-Anweisung zu erstellen, nehmen Sie die `cassandra:Create` Aktion in seine Richtlinie auf. Richtlinienanweisungen müssen entweder ein `Action` oder ein `NotAction`-Element enthalten. Amazon Keyspaces definiert eigene Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service ausführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": [
 "cassandra:CREATE",
 "cassandra:MODIFY"
]
```

Eine Liste der Amazon Keyspaces-Aktionen finden Sie unter [Von Amazon Keyspaces definierte Aktionen \(für Apache Cassandra\)](#) in der Service Authorization Reference.

## Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, 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": "*"
```

In Amazon Keyspaces können Schlüsselräume und Tabellen im `Resource` Element der IAM-Berechtigungen verwendet werden.

Die Amazon Keyspace-Ressource hat den folgenden ARN:

```
arn:${Partition}:cassandra:${Region}:${Account}:/keyspace/${KeyspaceName}/
```

Die Amazon Keyspaces-Tabellenressource hat den folgenden ARN:

```
arn:${Partition}:cassandra:${Region}:${Account}:/keyspace/${KeyspaceName}/table/${tableName}
```

Weitere Informationen zum Format von ARNs finden Sie unter [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#).

Um beispielsweise den `mykeyspace` Schlüsselraum in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN:

```
"Resource": "arn:aws:cassandra:us-east-1:123456789012:/keyspace/mykeyspace/"
```

Um alle Schlüsselräume anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (\*):



```
"Resource": "arn:aws:cassandra:us-east-1:123456789012:/keyspace/*"
```

Einige Amazon Keyspaces-Aktionen, z. B. zum Erstellen von Ressourcen, können nicht für eine bestimmte Ressource ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (\*) verwenden.

```
"Resource": "*"
```

Um programmgesteuert mit einem Standardtreiber eine Verbindung zu Amazon Keyspaces herzustellen, muss ein Principal SELECT-Zugriff auf die Systemtabellen haben, da die meisten Treiber die Systemschlüsselbereiche/Tabellen bei der Verbindung lesen. Um einem IAM-Benutzer beispielsweise SELECT Berechtigungen für `mytable` in zu gewähren, muss der Principal sowohl über `mykeyspace` Leseberechtigungen als auch für `system keyspace` verfügen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs durch Kommata voneinander.

```
"Resource": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
```

Eine Liste der Amazon Keyspaces-Ressourcentypen und ihrer ARNs finden Sie unter [Von Amazon Keyspaces definierte Ressourcen \(für Apache Cassandra\)](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von Amazon Keyspaces definierte Aktionen \(für Apache Cassandra\)](#).

## Bedingungsschlüssel

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, 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 dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Amazon Keyspaces definiert seinen eigenen Satz von Bedingungsschlüsseln und unterstützt auch die Verwendung einiger globaler Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontext-Schlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Alle Amazon Keyspaces-Aktionen unterstützen die Schlüssel `aws:RequestTag/${TagKey}` und `aws:ResourceTag/${TagKey}`, die und die `aws:TagKeys` Bedingungstasten. Weitere Informationen finden Sie unter [the section called “ Amazon Keyspaces-Ressourcenzugriff basierend auf Tags”](#).

Eine Liste der Amazon Keyspaces-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon Keyspaces \(für Apache Cassandra\)](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon Keyspaces definierte Aktionen \(für Apache Cassandra\)](#).

## Beispiele

Beispiele für identitätsbasierte Richtlinien von Amazon Keyspaces finden Sie unter [Beispiele für identitätsbasierte Richtlinien von Amazon Keyspaces](#)

## Ressourcenbasierte Richtlinien von Amazon Keyspaces

Amazon Keyspaces unterstützt keine ressourcenbasierten Richtlinien. Ein Beispiel für eine detaillierte Seite zu ressourcenbasierten Richtlinien finden Sie unter <https://docs.aws.amazon.com/lambda/latest/dg/access-control-resource-based.html>.

## Autorisierung basierend auf Amazon Keyspaces-Tags

Sie können den Zugriff auf Ihre Amazon Keyspaces-Ressourcen mithilfe von Tags verwalten. Um den Ressourcenzugriff auf der Grundlage von Tags zu verwalten, geben Sie Tag-Informationen im [Bedingungelement](#) einer Richtlinie mithilfe der `aws:TagKeys` Bedingungsschlüssel `cassandra:ResourceTag/key-name` `aws:RequestTag/key-name`, oder ein. Weitere Informationen zum Taggen von Amazon Keyspaces-Ressourcen finden Sie unter [the section called “Arbeiten mit Tags”](#)

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter [Amazon Keyspaces-Ressourcenzugriff basierend auf Tags](#).

## IAM-Rollen in Amazon Keyspaces

Eine [IAM-Rolle](#) ist eine Entität innerhalb von Ihrem AWS-Konto, die über bestimmte Berechtigungen verfügt.

### Temporäre Anmeldeinformationen mit Amazon Keyspaces verwenden

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie [AssumeRole](#) oder [GetFederationToken](#) aufrufen.

Amazon Keyspaces unterstützt die Verwendung temporärer Anmeldeinformationen mit dem AWS Signature Version 4 (Sigv4) -Authentifizierungs-Plugin, das im Github-Repo für die folgenden Sprachen verfügbar ist:

- Java: <https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin>
- Node.js: <https://github.com/aws/aws-sigv4-auth-cassandra-nodejs-driver-plugin>.
- Python: <https://github.com/aws/aws-sigv4-auth-cassandra-python-driver-plugin>.
- Geh: <https://github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin>.

Beispiele und Tutorials, die das Authentifizierungs-Plugin für den programmgesteuerten Zugriff auf Amazon Keyspaces implementieren, finden Sie unter [the section called “Verwenden eines Cassandra-Client-Treibers”](#)

## Service-verknüpfte Rollen

Mit [dienstbezogenen Rollen](#) können AWS Dienste auf Ressourcen in anderen Diensten zugreifen, um eine Aktion in Ihrem Namen durchzuführen. Serviceverknüpfte Rollen werden in Ihrem IAM-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von serviceverknüpften Amazon Keyspaces-Rollen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#)

## Service rollen

Amazon Keyspaces unterstützt keine Service rollen.

## Beispiele für identitätsbasierte Richtlinien von Amazon Keyspaces

Standardmäßig sind IAM-Benutzer und -Rollen nicht berechtigt, Amazon Keyspaces-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben über die Konsole, CQLSH oder API ausführen. AWS CLI AWS Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von Richtlinien auf der JSON-Registerkarte](#) im IAM-Benutzerhandbuch.

## Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der Amazon Keyspaces-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Zugreifen auf Amazon Keyspaces-Tabellen](#)
- [Amazon Keyspaces-Ressourcenzugriff basierend auf Tags](#)

## Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon Keyspaces-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen

AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu 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 speziell auf Ihre Anwendungsfälle zugeschnitten 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 Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. 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 [Richtlinienvvalidierung 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 System 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 Keyspaces-Konsole

Amazon Keyspaces benötigt keine speziellen Berechtigungen für den Zugriff auf die Amazon Keyspaces-Konsole. Sie benötigen mindestens Leseberechtigungen, um Details zu den Amazon Keyspaces-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 (IAM-Benutzer oder -Rollen) mit dieser Richtlinie.

Den Entitäten stehen zwei AWS verwaltete Richtlinien für den Zugriff auf die Amazon Keyspaces-Konsole zur Verfügung.

- [AmazonKeyspacesReadOnlyAccess\\_v2](#) — Diese Richtlinie gewährt nur Lesezugriff auf Amazon Keyspaces.
- [AmazonKeyspacesFullAccess](#) — Diese Richtlinie gewährt Berechtigungen zur Nutzung von Amazon Keyspaces mit vollem Zugriff auf alle Funktionen.

Weitere Informationen zu den verwalteten Richtlinien von Amazon Keyspaces finden Sie unter [the section called "Von AWS verwaltete Richtlinien"](#).

## 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 umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI API oder der AWS

```
{
 "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": "*"
 }
]
}

```

## Zugreifen auf Amazon Keyspaces-Tabellen

Im Folgenden finden Sie eine Beispielrichtlinie, die nur Lesezugriff (SELECT) auf die Amazon Keyspaces-Systemtabellen gewährt. Ersetzen Sie für alle Beispiele die Region und die Konto-ID im Amazon-Ressourcennamen (ARN) durch Ihre eigenen.

### Note

Um eine Verbindung mit einem Standardtreiber herzustellen, muss ein Benutzer mindestens SELECT Zugriff auf die Systemtabellen haben, da die meisten Treiber die Systemschlüsselbereiche/Tabellen bei der Verbindung lesen.

```

{
 "Version": "2012-10-17",

```

```

"Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "cassandra:Select"
],
 "Resource":[
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}

```

Die folgende Beispielrichtlinie fügt schreibgeschützten Zugriff auf die Benutzertabelle im Schlüsselraum hinzu. `mytable mykeyspace`

```

{
 "Version":"2012-10-17",
 "Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "cassandra:Select"
],
 "Resource":[
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}

```

Die folgende Beispielrichtlinie weist Lese-/Schreibzugriff auf eine Benutzertabelle und Lesezugriff auf die Systemtabellen zu.

#### Note

Systemtabellen sind immer schreibgeschützt.

```
{
```

```

"Version":"2012-10-17",
"Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource":[
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}

```

Die folgende Beispielrichtlinie ermöglicht es einem Benutzer, Tabellen im Schlüsselraum zu erstellen.  
mykeyspace

```

{
 "Version":"2012-10-17",
 "Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "cassandra:Create",
 "cassandra:Select"
],
 "Resource":[
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}

```

## Amazon Keyspaces-Ressourcenzugriff basierend auf Tags

Sie können Bedingungen in Ihrer identitätsbasierten Richtlinie verwenden, um den Zugriff auf Amazon Keyspaces-Ressourcen anhand von Tags zu steuern. Diese Richtlinien steuern die Sichtbarkeit der Schlüsselbereiche und Tabellen im Konto. Beachten Sie, dass sich tagbasierte



Berechtigungen für Systemtabellen anders verhalten, wenn Anfragen über das AWS SDK gestellt werden, als bei API-Aufrufen der Cassandra Query Language (CQL) über Cassandra-Treiber und Entwicklertools.

- Um Anfragen mit dem AWS SDK stellen `List` und `Get` Ressourcen bereitstellen zu können, wenn der Tag-basierte Zugriff verwendet wird, benötigt der Aufrufer Lesezugriff auf Systemtabellen. Beispielsweise sind `Select` Aktionsberechtigungen erforderlich, um mithilfe der Operation Daten aus Systemtabellen zu lesen. `GetTable` Wenn der Aufrufer nur Tag-basierten Zugriff auf eine bestimmte Tabelle hat, schlägt ein Vorgang fehl, der zusätzlichen Zugriff auf eine Systemtabelle erfordert.
- Aus Gründen der Kompatibilität mit dem etablierten Cassandra-Treiberverhalten werden tagbasierte Autorisierungsrichtlinien nicht durchgesetzt, wenn Operationen an Systemtabellen mithilfe von Cassandra Query Language (CQL) -API-Aufrufen über Cassandra-Treiber und Entwicklertools ausgeführt werden.

Das folgende Beispiel zeigt, wie Sie eine Richtlinie erstellen können, die einem Benutzer Berechtigungen zum Anzeigen einer Tabelle gewährt, wenn die Tabelle den Wert des Benutzernamens dieses Benutzers `Owner` enthält. In diesem Beispiel gewähren Sie auch Lesezugriff auf die Systemtabellen.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ReadOnlyAccessTaggedTables",
 "Effect": "Allow",
 "Action": "cassandra:Select",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
],
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Owner": "${aws:username}"
 }
 }
 }
]
}
```

Sie können diese Richtlinie den IAM-Benutzern in Ihrem Konto anfügen. Wenn ein benannter Benutzer `richard-roe` versucht, eine Amazon Keyspaces-Tabelle aufzurufen, muss die Tabelle mit `Owner=richard-roe` oder `owner=richard-roe` gekennzeichnet sein. Andernfalls wird der Zugriff abgelehnt. Der Tag-Schlüssel `Owner` der Bedingung stimmt sowohl mit `Owner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

Die folgende Richtlinie gewährt einem Benutzer die Erlaubnis, Tabellen mit Tags zu erstellen, wenn die Tabelle den Wert des Benutzernamens dieses Benutzers `Owner` enthält.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateTagTableUser",
 "Effect": "Allow",
 "Action": [
 "cassandra:Create",
 "cassandra:TagResource"
],
 "Resource": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Owner": "${aws:username}"
 }
 }
 }
]
}
```

## AWSverwaltete Richtlinien für Amazon Keyspaces

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Services verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

## AWSverwaltete Richtlinie: `_v2 AmazonKeyspacesReadOnlyAccess`

Sie können die `AmazonKeyspacesReadOnlyAccess_v2`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt nur Lesezugriff auf Amazon Keyspaces und beinhaltet die erforderlichen Berechtigungen, wenn eine Verbindung über private VPC-Endpunkte hergestellt wird.

### Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `Amazon Keyspaces`— Bietet schreibgeschützten Zugriff auf Amazon Keyspaces.
- `Application Auto Scaling`— Ermöglicht Prinzipalen das Anzeigen von Konfigurationen aus Application Auto Scaling. Dies ist erforderlich, damit Benutzer automatische Skalierungsrichtlinien anzeigen können, die an eine Tabelle angehängt sind.
- `CloudWatch`— Ermöglicht Prinzipalen das Anzeigen von Metrikdaten und Alarmen, die in CloudWatch konfiguriert sind. Dies ist erforderlich, damit Benutzer die Größe der Fakturierungstabelle und die CloudWatch Alarme, die für eine Tabelle konfiguriert wurden, einsehen können.

- **AWS KMS**— Ermöglicht es den Hauptbenutzern, die in konfigurierten Schlüssel anzuzeigen. **AWS KMS** Dies ist erforderlich, damit Benutzer die AWS KMS Schlüssel einsehen können, die sie in ihrem Konto erstellen und verwalten, um zu bestätigen, dass es sich bei dem Amazon Keyspaces zugewiesenen Schlüssel um einen aktivierten symmetrischen Verschlüsselungsschlüssel handelt.
- **Amazon EC2**— Ermöglicht Principals, die sich über VPC-Endpunkte mit Amazon Keyspaces verbinden, die VPC auf Ihrer Amazon EC2 EC2-Instance nach Endpunkt- und Netzwerkschnittstelleninformationen abzufragen. Dieser schreibgeschützte Zugriff auf die Amazon EC2 EC2-Instance ist erforderlich, damit Amazon Keyspaces verfügbare Schnittstellen-VPC-Endpunkte in der `system.peers` Tabelle suchen und speichern kann, die für den Verbindungslastenausgleich verwendet wird.

[Um die Richtlinie im Format zu überprüfen, siehe `\_v2.JSONAmazonKeyspacesReadOnlyAccess`](#)

## AWSverwaltete Richtlinie: `AmazonKeyspacesReadOnlyAccess`

Sie können die `AmazonKeyspacesReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt nur Lesezugriff auf Amazon Keyspaces.

### Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- **Amazon Keyspaces**— Bietet schreibgeschützten Zugriff auf Amazon Keyspaces.
- **Application Auto Scaling**— Ermöglicht Prinzipalen das Anzeigen von Konfigurationen aus Application Auto Scaling. Dies ist erforderlich, damit Benutzer automatische Skalierungsrichtlinien anzeigen können, die an eine Tabelle angehängt sind.
- **CloudWatch**— Ermöglicht Prinzipalen das Anzeigen von Metrikdaten und Alarmen, die in CloudWatch konfiguriert sind. Dies ist erforderlich, damit Benutzer die Größe der Fakturierungstabelle und die CloudWatch Alarme, die für eine Tabelle konfiguriert wurden, einsehen können.
- **AWS KMS**— Ermöglicht es den Hauptbenutzern, die in konfigurierten Schlüssel anzuzeigen. **AWS KMS** Dies ist erforderlich, damit Benutzer die AWS KMS Schlüssel einsehen können, die sie in

ihrem Konto erstellen und verwalten, um zu bestätigen, dass es sich bei dem Amazon Keyspaces zugewiesenen Schlüssel um einen aktivierten symmetrischen Verschlüsselungsschlüssel handelt.

Informationen zur JSON Formatierung der Richtlinie finden Sie unter.

[AmazonKeyspacesReadOnlyAccess](#)

## AWS verwaltete Richtlinie: AmazonKeyspacesFullAccess

Sie können die AmazonKeyspacesFullAccess-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Administratorberechtigungen, die Ihren Administratoren uneingeschränkten Zugriff auf Amazon Keyspaces ermöglichen.

### Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- **Amazon Keyspaces**— Ermöglicht Prinzipalen den Zugriff auf jede Amazon Keyspaces-Ressource und die Ausführung aller Aktionen.
- **Application Auto Scaling**— Ermöglicht Prinzipalen das Erstellen, Anzeigen und Löschen automatischer Skalierungsrichtlinien für Amazon Keyspaces-Tabellen. Dies ist erforderlich, damit Administratoren automatische Skalierungsrichtlinien für Amazon Keyspaces-Tabellen verwalten können.
- **CloudWatch**— Ermöglicht Prinzipalen, die Größe der fakturierbaren Tabelle zu sehen sowie CloudWatch Alarme für automatische Skalierungsrichtlinien von Amazon Keyspace zu erstellen, anzuzeigen und zu löschen. Dies ist erforderlich, damit Administratoren die Größe der fakturierbaren Tabelle einsehen und ein Dashboard erstellen können. CloudWatch
- **IAM**— Ermöglicht Amazon Keyspaces, automatisch serviceverknüpfte Rollen mit IAM zu erstellen, wenn die folgenden Funktionen aktiviert sind:
  - **Application Auto Scaling**— Wenn ein Administrator Application Auto Scaling für eine Tabelle aktiviert, erstellt Amazon Keyspaces eine serviceverknüpfte Rolle, um automatische Skalierungsaktionen in Ihrem Namen durchzuführen.
  - **Amazon Keyspaces Multi-Region Replication**— Wenn ein Administrator einen Schlüsselraum für mehrere Regionen erstellt, wird automatisch eine servicebezogene Rolle

erstellt, um die Datenreplikation für den ausgewählten Schlüsselraum in Ihrem Namen durchzuführen. AWS-Regionen

Weitere Informationen zu Service-verknüpften Rollen finden Sie unter [the section called “Verwenden von serviceverknüpften Rollen”](#).

- **AWS KMS**— Ermöglicht Prinzipalen das Anzeigen von Schlüsseln, die in konfiguriert sind. AWS KMS Dies ist erforderlich, damit Benutzer die AWS KMS Schlüssel einsehen können, die sie in ihrem Konto erstellen und verwalten, um zu bestätigen, dass es sich bei dem Amazon Keyspaces zugewiesenen Schlüssel um einen aktivierten symmetrischen Verschlüsselungsschlüssel handelt.
- **Amazon EC2**— Ermöglicht Principals, die sich über VPC-Endpunkte mit Amazon Keyspaces verbinden, die VPC auf Ihrer Amazon EC2 EC2-Instance nach Endpunkt- und Netzwerkschnittstelleninformationen abzufragen. Dieser schreibgeschützte Zugriff auf die Amazon EC2 EC2-Instance ist erforderlich, damit Amazon Keyspaces verfügbare Schnittstellen-VPC-Endpunkte in der `system.peers` Tabelle suchen und speichern kann, die für den Verbindungslastenausgleich verwendet wird.

Informationen zur Formatierung der Richtlinie finden Sie unter. JSON [AmazonKeyspacesFullAccess](#)

## Amazon Keyspaces-Aktualisierungen der AWS verwalteten Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon Keyspaces an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der [Dokumentverlauf](#)-Seite.

| Änderung                                                                                     | Beschreibung                                                                                                                                                                          | Datum           |
|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">AmazonKeyspacesFullAccess</a><br>– Aktualisierung auf eine bestehende Richtlinie | Amazon Keyspaces hat neue Nur-Lese-Berechtigungen für Clients hinzugefügt, die sich über Schnittstellen-VPC-Endpunkte mit Amazon Keyspaces verbinden, um auf die Amazon EC2 EC2-Insta | 3. Oktober 2023 |

| Änderung                                                                  | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Datum                     |
|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
|                                                                           | <p>nce zuzugreifen, um Netzwerki<br/>nformationen abzurufen.</p> <p>Amazon Keyspaces speichert verfügbare VPC-Schni<br/>ttstellen-Endpunkte in der <code>system.peers</code> Tabelle für den Verbindungslastena<br/>usgleich. Weitere Informati<br/>onen finden Sie unter <a href="#">the section called “Verwendu<br/>ng von -Schnittstellen-VPC-<br/>Endpunkten”</a>.</p>                                                                                                                                                                                                                         |                           |
| <p><a href="#">AmazonKeyspacesReadOnlyAccess_v2</a> — Neue Richtlinie</p> | <p>Amazon Keyspaces hat eine neue Richtlinie erstellt, um Nur-Lese-Berechtigungen für Clients hinzuzufügen, die sich über VPC-Schnittstellen-Endpunkte mit Amazon Keyspaces verbinden, um auf die Amazon EC2 EC2-Insta<br/>nce zuzugreifen, um Netzwerki<br/>nformationen abzurufen.</p> <p>Amazon Keyspaces speichert verfügbare VPC-Schni<br/>ttstellen-Endpunkte in der <code>system.peers</code> Tabelle für den Verbindungslastena<br/>usgleich. Weitere Informati<br/>onen finden Sie unter <a href="#">the section called “Verwendu<br/>ng von -Schnittstellen-VPC-<br/>Endpunkten”</a>.</p> | <p>12. September 2023</p> |

| Änderung                                                                                                 | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                            | Datum               |
|----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <p><a href="#">AmazonKeyspacesFullAccess</a><br/>– Aktualisierung auf eine bestehende Richtlinie</p>     | <p>Amazon Keyspaces hat neue Berechtigungen hinzugefügt, die es Amazon Keyspaces ermöglichen, eine servicebezogene Rolle zu erstellen, wenn ein Administrator einen Schlüsselraum mit mehreren Regionen erstellt.</p> <p>Amazon Keyspaces verwendet die serviceverknüpfte Rolle, um Datenreplikationsaufgaben in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter <a href="#">the section called “Multiregionale Replikation”</a>.</p> | <p>5. Juni 2023</p> |
| <p><a href="#">AmazonKeyspacesReadOnlyAccess</a><br/>– Aktualisierung auf eine bestehende Richtlinie</p> | <p>Amazon Keyspaces hat neue Berechtigungen hinzugefügt, mit denen Benutzer die abrechnungsfähige Größe einer Tabelle mithilfe von CloudWatch anzeigen können.</p> <p>Amazon Keyspaces ist in Amazon integriert CloudWatch, sodass Sie die Größe der fakturierbaren Tabelle überwachen können. Weitere Informationen finden Sie unter <a href="#">the section called “Amazon Keyspaces-Metriken und Dimensionen”</a>.</p>                               | <p>7. Juli 2022</p> |



| Änderung                                                                                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                               | Datum        |
|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <a href="#">AmazonKeyspacesFullAccess</a><br>– Aktualisierung auf eine bestehende Richtlinie | <p>Amazon Keyspaces hat neue Berechtigungen hinzugefügt, mit denen Benutzer die abrechnungsfähige Größe einer Tabelle mithilfe von anzeigen können. CloudWatch</p> <p>Amazon Keyspaces ist in Amazon integriert CloudWatch , sodass Sie die Größe der fakturierbaren Tabelle überwachen können. Weitere Informationen finden Sie unter <a href="#">the section called “Amazon Keyspaces-Metriken und Dimensionen”</a>.</p> | 7. Juli 2022 |

| Änderung                                                                                      | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Datum        |
|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <a href="#">AmazonKeyspacesReadOnlyAccess</a> – Aktualisierung auf eine bestehende Richtlinie | <p>Amazon Keyspaces hat neue Berechtigungen hinzugefügt, um Benutzern das Anzeigen von AWS KMS Schlüsseln zu ermöglichen, die für die Amazon Keyspaces-Verschlüsselung im Ruhezustand konfiguriert wurden.</p> <p>Amazon Keyspaces Encryption at Rest lässt sich integrieren, um die Verschlüsselungsschlüssel zu schützen und zu verwalten, die zur Verschlüsselung von Daten im Ruhezustand verwendet werden. AWS KMS Um den für Amazon Keyspaces konfigurierten AWS KMS Schlüssel anzuzeigen, wurden Nur-Lese-Berechtigungen hinzugefügt.</p> | 1. Juni 2021 |

| Änderung                                                                                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Datum        |
|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <a href="#">AmazonKeyspacesFullAccess</a><br>– Aktualisierung auf eine bestehende Richtlinie | <p>Amazon Keyspaces hat neue Berechtigungen hinzugefügt, um Benutzern das Anzeigen von AWS KMS Schlüsseln zu ermöglichen, die für die Amazon Keyspaces-Verschlüsselung im Ruhezustand konfiguriert wurden.</p> <p>Amazon Keyspaces Encryption at Rest lässt sich integrieren, um die Verschlüsselungsschlüssel zu schützen und zu verwalten, die zur Verschlüsselung von Daten im Ruhezustand verwendet werden. AWS KMS Um den für Amazon Keyspaces konfigurierten AWS KMS Schlüssel anzuzeigen, wurden Nur-Lese-Berechtigungen hinzugefügt.</p> | 1. Juni 2021 |
| Amazon Keyspaces hat begonnen, Änderungen zu verfolgen                                       | Amazon Keyspaces hat damit begonnen, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1. Juni 2021 |

## Fehlerbehebung bei Identität und Zugriff auf Amazon Keyspaces

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Keyspaces und IAM auftreten können.

### Themen

- [Ich bin nicht berechtigt, eine Aktion in Amazon Keyspaces durchzuführen](#)

- [Ich habe einen IAM-Benutzer oder eine IAM-Rolle geändert und die Änderungen wurden nicht sofort wirksam](#)
- [Ich kann eine Tabelle mit Amazon Keyspaces point-in-time Recovery \(PITR\) nicht wiederherstellen](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich bin Administrator und möchte anderen den Zugriff auf Amazon Keyspaces ermöglichen](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon Keyspaces-Ressourcen ermöglichen](#)

## Ich bin nicht berechtigt, eine Aktion in Amazon Keyspaces durchzuführen

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM-Benutzer versucht, die Konsole zu verwenden, um Details zu einer *Tabelle* anzuzeigen, aber keine `cassandra:Select` Berechtigungen für die Tabelle besitzt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
cassandra:Select on resource: mytable
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *mytable* auf die Ressource `cassandra:Select` zugreifen zu können.

## Ich habe einen IAM-Benutzer oder eine IAM-Rolle geändert und die Änderungen wurden nicht sofort wirksam

Es kann bis zu 10 Minuten dauern, bis Änderungen der IAM-Richtlinien für Anwendungen mit bestehenden, etablierten Verbindungen zu Amazon Keyspaces wirksam werden. Änderungen der IAM-Richtlinien werden sofort wirksam, wenn Anwendungen eine neue Verbindung herstellen. Wenn Sie Änderungen an einem vorhandenen IAM-Benutzer oder einer vorhandenen IAM-Rolle vorgenommen haben und diese nicht sofort wirksam wurden, warten Sie entweder 10 Minuten oder trennen Sie die Verbindung und stellen Sie erneut eine Verbindung zu Amazon Keyspaces her.

## Ich kann eine Tabelle mit Amazon Keyspaces point-in-time Recovery (PITR) nicht wiederherstellen

Wenn Sie versuchen, eine Amazon Keyspaces-Tabelle mit point-in-time Wiederherstellung (PITR) wiederherzustellen, und Sie sehen, dass der Wiederherstellungsvorgang beginnt, aber nicht erfolgreich abgeschlossen wird, haben Sie möglicherweise nicht alle erforderlichen Berechtigungen konfiguriert, die für den Wiederherstellungsvorgang erforderlich sind. Sie müssen sich an Ihren Administrator wenden, um Unterstützung zu erhalten, und diese Person bitten, Ihre Richtlinien zu aktualisieren, damit Sie eine Tabelle in Amazon Keyspaces wiederherstellen können.

Zusätzlich zu den Benutzerberechtigungen benötigt Amazon Keyspaces möglicherweise Berechtigungen, um während des Wiederherstellungsvorgangs im Namen Ihres Auftraggebers Aktionen durchzuführen. Dies ist der Fall, wenn die Tabelle mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist oder wenn Sie IAM-Richtlinien verwenden, die den eingehenden Datenverkehr einschränken. Wenn Sie beispielsweise Bedingungsschlüssel in Ihrer IAM-Richtlinie verwenden, um den Quelldatenverkehr auf bestimmte Endpunkte oder IP-Bereiche zu beschränken, schlägt der Wiederherstellungsvorgang fehl. Damit Amazon Keyspaces die Tabellenwiederherstellung im Namen Ihres Prinzipals durchführen kann, müssen Sie der IAM-Richtlinie einen `aws:ViaAWSService` globalen Bedingungsschlüssel hinzufügen.

Weitere Informationen zu Berechtigungen zum Wiederherstellen von Tabellen finden Sie unter [the section called “Berechtigungen wiederherstellen”](#)

## Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon Keyspaces übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene 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 Keyspaces 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 bin Administrator und möchte anderen den Zugriff auf Amazon Keyspaces ermöglichen

Um anderen den Zugriff auf Amazon Keyspaces zu ermöglichen, müssen Sie eine IAM-Entität (Benutzer oder Rolle) für die Person oder Anwendung erstellen, die Zugriff benötigt. Sie werden die Anmeldeinformationen für diese Einrichtung verwenden, um auf AWS zuzugreifen. Anschließend müssen Sie der Entität eine Richtlinie beifügen, die ihr die richtigen Berechtigungen in Amazon Keyspaces gewährt.

Informationen zum Einstieg finden Sie unter [Erstellen Ihrer ersten delegierten IAM-Benutzer und -Gruppen](#) im IAM-Benutzerhandbuch.

## Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon Keyspaces-Ressourcen ermöglichen

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 Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (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 darüber, ob Amazon Keyspaces diese Funktionen unterstützt, finden Sie unter [So funktioniert Amazon Keyspaces mit IAM](#).
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto, den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).

- 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 im IAM-Benutzerhandbuch unter [Kontenübergreifender Ressourcenzugriff in IAM](#).

## Verwenden von serviceverknüpften Rollen für Amazon Keyspaces

Amazon Keyspaces (für Apache Cassandra) verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Amazon Keyspaces verknüpft ist. Serviceverknüpfte Rollen werden von Amazon Keyspaces vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer -AWSServices in Ihrem Namen erfordert.

### Themen

- [Verwenden von Rollen für Amazon Keyspaces Application Auto Scaling](#)
- [Verwenden von Rollen für die Multi-Region-Replikation in Amazon Keyspaces](#)

## Verwenden von Rollen für Amazon Keyspaces Application Auto Scaling

Amazon Keyspaces (für Apache Cassandra) verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Amazon Keyspaces verknüpft ist. Serviceverknüpfte Rollen werden von Amazon Keyspaces vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer -AWSServices in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht das Einrichten von Amazon Keyspaces, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon Keyspaces definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, können nur Amazon Keyspaces die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen juristischen Stelle von IAM zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Dies schützt Ihre Amazon Keyspaces-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceorientierte Rollen unterstützen, finden Sie unter [AWS services that work with IAM](#) (-Services, die mit IAM funktionieren). Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Service-linked roles (Serviceorientierte Rollen) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

## Berechtigungen von serviceverknüpften Rollen für Amazon Keyspaces

Amazon Keyspaces verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForApplicationAutoScaling_CassandraTable`, damit Application Auto Scaling Amazon Keyspaces und Amazon CloudWatch in Ihrem Namen aufrufen kann.

Die `AWSServiceRoleForApplicationAutoScaling_CassandraTable` serviceverknüpfte Rolle vertraut darauf, dass die folgenden Services die Rolle übernehmen:

- `cassandra.application-autoscaling.amazonaws.com`

Die Rollenberechtigungsrichtlinie ermöglicht Application Auto Scaling die Durchführung der folgenden Aktionen für die angegebenen Amazon Keyspaces-Ressourcen:


- Aktion: `cassandra:Select` für `arn:*:cassandra:*:*:/keyspace/system/table/*`
- Aktion: `cassandra:Select` auf die Ressource `arn:*:cassandra:*:*:/keyspace/system_schema/table/*`
- Aktion: `cassandra:Select` auf die Ressource `arn:*:cassandra:*:*:/keyspace/system_schema_mcs/table/*`
- Aktion: `cassandra:Alter` auf Ressource `arn:*:cassandra:*:*:"*`

## Erstellen einer serviceverknüpften Rolle für Amazon Keyspaces

Sie müssen keine serviceverknüpfte Rolle für die automatische Skalierung von Amazon Keyspaces manuell erstellen. Wenn Sie die automatische Skalierung von Amazon Keyspaces für eine Tabelle mit der AWS Management Console, dem CQLAWS CLI, der oder der AWS API aktivieren, erstellt Application Auto Scaling die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie Amazon Keyspaces Auto Scaling für eine Tabelle aktivieren, erstellt Application Auto Scaling die serviceverknüpfte Rolle erneut für Sie.



 **Important**

Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Weitere Informationen finden Sie unter [In meinem AWS-Konto](#) wird eine neue Rolle angezeigt.


Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie die automatische Anwendungsskalierung von Amazon Keyspaces für eine Tabelle aktivieren, erstellt Application Auto Scaling die serviceverknüpfte Rolle erneut für Sie.

### Bearbeiten einer serviceverknüpften Rolle für Amazon Keyspaces

Amazon Keyspaces erlaubt es Ihnen nicht, die `AWSServiceRoleForApplicationAutoScaling_CassandraTable` serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename 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 IAM-Benutzerhandbuch.

### Löschen einer serviceverknüpften Rolle für Amazon Keyspaces

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 ist keine ungenutzte Entität vorhanden, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch zuerst die automatische Skalierung für alle Tabellen im Konto für alle deaktivieren, AWS-Regionen bevor Sie die serviceverknüpfte Rolle manuell löschen können. Informationen zum Deaktivieren der automatischen Skalierung für Amazon-Keyspaces-Tabellen finden Sie unter [Ändern oder Deaktivieren der automatischen Skalierungseinstellungen für Amazon Keyspaces](#).

 **Note**

Wenn Amazon Keyspaces Auto Scaling die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu ändern, schlägt die Abmeldung möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

## So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die `-AWSAPIAWS CLI`, um die `AWSServiceRoleForApplicationAutoScaling_CassandraTable` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Leitfaden.

### Note

Um die serviceverknüpfte Rolle zu löschen, die von Amazon Keyspaces Auto Scaling verwendet wird, müssen Sie zuerst die automatische Skalierung für alle Tabellen im Konto deaktivieren.

## Unterstützte Regionen für serviceverknüpfte Amazon Keyspaces-Rollen

Amazon Keyspaces unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [Service-Endpunkte für Amazon Keyspaces](#).

## Verwenden von Rollen für die Multi-Region-Replikation in Amazon Keyspaces

Amazon Keyspaces (für Apache Cassandra) verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Amazon Keyspaces verknüpft ist. Serviceverknüpfte Rollen werden von Amazon Keyspaces vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer `-AWSServices` in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht das Einrichten von Amazon Keyspaces, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon Keyspaces definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, können nur Amazon Keyspaces die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen juristischen Stelle von IAM zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Dies schützt Ihre Amazon Keyspaces-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceorientierte Rollen unterstützen, finden Sie unter [AWS services that work with IAM](#) (`-Services`, die mit IAM funktionieren). Suchen Sie nach den Services, für

die Yes (Ja) in der Spalte Service-linked roles (Serviceorientierte Rollen) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

## Berechtigungen von serviceverknüpften Rollen für Amazon Keyspaces

Amazon Keyspaces verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonKeyspacesReplication`, um Amazon Keyspaces zu erlauben, Schreibvorgänge in alle Replikate einer multiregionalen Tabelle in Ihrem Namen zu replizieren.

Die `AWSServiceRoleForAmazonKeyspacesReplication` serviceverknüpfte Rolle vertraut darauf, dass die folgenden Services die Rolle übernehmen:

- `replication.cassandra.amazonaws.com`

Die Rollenberechtigungsrichtlinie namens `KeyspacesReplicationServiceRolePolicy` erlaubt Amazon Keyspaces, die folgenden Aktionen durchzuführen:

- Aktion: `cassandra:Select`
- Aktion: `cassandra:SelectMultiRegionResource`
- Aktion: `cassandra:Modify`
- Aktion: `cassandra:ModifyMultiRegionResource`

Obwohl die serviceverknüpfte Rolle von Amazon Keyspaces die Berechtigungen „Aktion:“ für den angegebenen Amazon-Ressourcennamen (ARN) „arn:\*“ in der Richtlinie `AWSServiceRoleForAmazonKeyspacesReplication` bereitstellt, stellt Amazon Keyspaces den ARN Ihres Kontos bereit.

Sie müssen Berechtigungen konfigurieren, damit eine Benutzer, Gruppen oder Rollen eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter [serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

## Erstellen einer serviceverknüpften Rolle für Amazon Keyspaces

Sie können eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen multiregionalen Keyspace in der AWS CLI, AWS Management Console oder der -AWSAPI erstellen, erstellt Amazon Keyspaces die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen multiregionalen Keyspace erstellen, erstellt Amazon Keyspaces die serviceverknüpfte Rolle erneut für Sie.

### Bearbeiten einer serviceverknüpften Rolle für Amazon Keyspaces

Amazon Keyspaces erlaubt es Ihnen nicht, die `AWSServiceRoleForAmazonKeyspacesReplication` 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 IAM-Benutzerhandbuch.

### Löschen einer serviceverknüpften Rolle für Amazon Keyspaces

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften 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 zunächst alle multiregionalen Keyspaces im Konto über alle hinweg löschen, AWS-Regionen bevor Sie die serviceverknüpfte Rolle manuell löschen können.

### Bereinigen einer serviceverknüpften Rolle

Bevor Sie IAM zum Löschen einer serviceverknüpften Rolle verwenden können, müssen Sie zunächst alle multiregionalen Keyspaces und Tabellen löschen, die von der Rolle verwendet werden.

#### Note

Wenn der Amazon Keyspaces-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie Amazon Keyspaces-Ressourcen, die von der `AWSServiceRoleForAmazonKeyspacesReplication` (Konsole) verwendet werden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Keyspaces-Konsole unter <https://console.aws.amazon.com/keyspaces/home>.
2. Wählen Sie im linken Bereich Keyspaces aus.

3. Wählen Sie alle multiregionalen Keyspaces aus der Liste aus.
4. Wählen Sie Löschen, bestätigen Sie den Löschvorgang und wählen Sie Keyspaces löschen aus.

Sie können multiregionale Keyspaces auch programmgesteuert mit einer der folgenden Methoden löschen.

- Die Cassandra Query Language (CQL)-[DELETE](#)-Anweisung.
- Die [delete-keyspace](#)-Operation der AWS CLI.
- Der [DeleteKeyspace](#) Betrieb der Amazon Keyspaces API.

### Manuelles Löschen der -serviceverknüpften Rolle

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die `AWSServiceRoleForAmazonKeyspacesReplication` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

### Unterstützte Regionen für serviceverknüpfte Amazon Keyspaces-Rollen

Amazon Keyspaces unterstützt nicht die Verwendung von serviceverknüpften Rollen in jeder Region, in der der Service verfügbar ist. Sie können die `AWSServiceRoleForAmazonKeyspacesReplication` Rolle in den folgenden Regionen verwenden.

| Name der Region            | Regions-ID     | Unterstützung in Amazon Keyspaces |
|----------------------------|----------------|-----------------------------------|
| USA Ost (Nord-Virginia)    | us-east-1      | Ja                                |
| USA Ost (Ohio)             | us-east-2      | Ja                                |
| USA West (Nordkalifornien) | us-west-1      | Ja                                |
| USA West (Oregon)          | us-west-2      | Ja                                |
| Asien-Pazifik (Mumbai)     | ap-south-1     | Ja                                |
| Asien-Pazifik (Osaka)      | ap-northeast-3 | Ja                                |
| Asien-Pazifik (Seoul)      | ap-northeast-2 | Ja                                |

| Name der Region           | Regions-ID     | Unterstützung in Amazon Keyspaces |
|---------------------------|----------------|-----------------------------------|
| Asien-Pazifik (Singapore) | ap-southeast-1 | Ja                                |
| Asien-Pazifik (Sydney)    | ap-southeast-2 | Ja                                |
| Asien-Pazifik (Tokyo)     | ap-northeast-1 | Ja                                |
| Kanada (Zentral)          | ca-central-1   | Ja                                |
| Europa (Frankfurt)        | eu-central-1   | Ja                                |
| Europa (Irland)           | eu-west-1      | Ja                                |
| Europa (London)           | eu-west-2      | Ja                                |
| Europa (Paris)            | eu-west-3      | Ja                                |
| Südamerika (São Paulo)    | sa-east-1      | Ja                                |
| AWS GovCloud (USA-Ost)    | us-gov-east-1  | Nein                              |
| AWS GovCloud (USA-West)   | us-gov-west-1  | Nein                              |

## Konformitätsprüfung für Amazon Keyspaces (für Apache Cassandra)

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon Keyspaces (für Apache Cassandra) im Rahmen mehrerer AWS Compliance-Programme. Dazu zählen:


- ISO/IEC 27001:2013, 27017:2015, 27018:2019 und ISO/IEC 9001:2015. Weitere Informationen finden Sie unter ISO- und CSA STAR-Zertifizierungen und [AWS -Services](#).
- System and Organization Controls (SOC)
- Payment Card Industry (PCI)
- Federal Risk and Authorization Management Program (FedRAMP) Hoch
- Health Insurance Portability and Accountability Act (HIPAA)

Informationen darüber, ob AWS-Service ein in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter [AWS-Services Umfang nach Compliance-Programm unter Umfang nach Compliance-Programm AWS-Services](#) das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-fähige Anwendungen erstellen AWS können.

 Note

AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- [AWS Security Hub](#)— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

## Ausfallsicherheit und Notfallwiederherstellung

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.

Amazon Keyspaces repliziert Daten automatisch dreimal in mehreren AWS Availability Zones innerhalb derselben, AWS-Region um Beständigkeit und hohe Verfügbarkeit zu gewährleisten.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Neben der AWS globalen Infrastruktur stellt Amazon Keyspaces verschiedene Funktionen bereit, um Ihren Anforderungen an Ausfallsicherheit und Datensicherung gerecht zu werden.

### Multiregionale Replikation

Amazon Keyspaces stellt Multiregionale Replikation bereit, falls Sie Ihre Daten oder Anwendungen über größere geografische Distanzen hinweg replizieren müssen. Sie können Ihre



Amazon Keyspaces-Tabellen auf bis zu sechs verschiedene Tabellen AWS-Regionen Ihrer Wahl replizieren. Weitere Informationen finden Sie unter [Multiregionale Replikation](#).

oint-in-timeZeitpunktbezogene Wiederherstellung

Mit der zeitpunktbezogenen Sicherung schützen Sie Ihre Amazon-Keyspaces-Tabellen vor versehentlichen Schreib- und Löschoptionen. Weitere Informationen finden Sie unter [oint-in-timeP-Wiederherstellung für Amazon Keyspaces](#).

## Infrastruktursicherheit in Amazon Keyspaces

Als verwalteter Service ist Amazon Keyspaces (für Apache Cassandra) durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter [AWS Cloud-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.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Keyspaces zuzugreifen. 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.

Amazon Keyspaces unterstützt zwei Methoden zur Authentifizierung von Kundenanfragen. Die erste Methode verwendet dienstspezifische Anmeldeinformationen, bei denen es sich um kennwortbasierte Anmeldeinformationen handelt, die für einen bestimmten IAM-Benutzer generiert wurden. Sie können das Passwort mithilfe der IAM-Konsole, der oder der AWS API erstellen AWS CLI und verwalten. Weitere Informationen finden Sie unter [Verwenden von IAM mit Amazon Keyspaces](#).

Die zweite Methode verwendet ein Authentifizierungs-Plugin für den DataStax Open-Source-Java-Treiber für Cassandra. Dieses Plugin ermöglicht es [IAM-Benutzern, Rollen und Verbundidentitäten](#), mithilfe des [AWSSignature Version 4-Prozesses](#) (SIGv4) Authentifizierungsinformationen zu Amazon

Keyspaces (für Apache Cassandra) API-Anfragen hinzuzufügen. Weitere Informationen finden Sie unter [the section called “IAM-Anmeldeinformationen für die Authentifizierung AWS”](#).

Sie können einen VPC-Schnittstellenendpunkt verwenden, um zu verhindern, dass der Datenverkehr zwischen Ihrer Amazon VPC und Amazon Keyspaces das Amazon-Netzwerk verlässt. Interface-VPC-Endpunkte werden von einer AWS Technologie unterstützt AWS PrivateLink, die private Kommunikation zwischen AWS Diensten mithilfe einer elastischen Netzwerkschnittstelle mit privaten IPs in Ihrer Amazon VPC ermöglicht. Weitere Informationen finden Sie unter [the section called “Verwendung von -Schnittstellen-VPC-Endpunkten”](#).

## Verwenden von Amazon Keyspaces mit Schnittstellen-VPC-Endpunkten

Schnittstellen-VPC-Endpunkte ermöglichen die private Kommunikation zwischen Ihrer Virtual Private Cloud (VPC), die in Amazon VPC läuft, und Amazon Keyspaces. Schnittstelle, über die VPC-Endpunkte betrieben werden. Dabei handelt es sich um einen AWS Dienst AWS PrivateLink, der die private Kommunikation zwischen VPCs und Diensten ermöglicht. AWS

AWS PrivateLink ermöglicht dies durch die Verwendung einer elastic network interface mit privaten IP-Adressen in Ihrer VPC, sodass der Netzwerkverkehr das Amazon-Netzwerk nicht verlässt. Schnittstellen-VPC-Endpunkte benötigen keinen Internet-Gateway, kein NAT-Gerät, keine VPN-Verbindung und keine AWS Direct Connect -Verbindung. Weitere Informationen finden Sie unter [Amazon Virtual Private Cloud](#) und [Interface VPC-Endpoints \(AWS PrivateLink\)](#).

### Themen

- [Verwenden von Schnittstellen-VPC-Endpunkten für Amazon Keyspaces](#)
- [Auffüllen von system.peers Tabelleneinträgen mit VPC-Endpunktinformationen der Schnittstelle](#)
- [Steuerung des Zugriffs auf VPC-Schnittstellen-Endpunkte für Amazon Keyspaces](#)
- [Verfügbarkeit](#)
- [VPC-Endpunkttrichtlinien und Amazon point-in-time Keyspaces-Wiederherstellung \(PITR\)](#)
- [Häufige Fehler und Warnungen](#)

## Verwenden von Schnittstellen-VPC-Endpunkten für Amazon Keyspaces

Sie können einen VPC-Schnittstellen-Endpunkt erstellen, sodass der Datenverkehr zwischen Amazon Keyspaces und Ihren Amazon VPC-Ressourcen über den VPC-Endpunkt der Schnittstelle fließt. Folgen Sie zunächst den Schritten zum [Erstellen eines](#) Schnittstellen-Endpunkts. Bearbeiten Sie als Nächstes die Sicherheitsgruppe, die dem Endpunkt zugeordnet ist, den Sie im vorherigen Schritt

erstellt haben, und konfigurieren Sie eine Regel für eingehenden Datenverkehr für Port 9142. Weitere Informationen finden Sie unter [Regeln hinzufügen, entfernen und aktualisieren](#).

Ein step-by-step Tutorial zur Konfiguration einer Verbindung zu Amazon Keyspaces über einen VPC-Endpunkt finden Sie unter [the section called “Verbindung mit VPC-Endpunkten herstellen”](#). Informationen zum Konfigurieren des kontoübergreifenden Zugriffs für Amazon Keyspaces-Ressourcen getrennt von Anwendungen AWS-Konten in verschiedenen VPC finden Sie unter [the section called “Kontenübergreifender Zugriff”](#)

## Auffüllen von **system.peers** Tabelleneinträgen mit VPC-Endpunktinformationen der Schnittstelle

Apache Cassandra-Treiber verwenden die `system.peers` Tabelle, um Knoteninformationen über den Cluster abzufragen. Cassandra-Treiber verwenden die Knoteninformationen für den Lastenausgleich bei Verbindungen und für Wiederholungsvorgänge. Amazon Keyspaces füllt automatisch neun Einträge in der `system.peers` Tabelle für Clients, die sich über den öffentlichen Endpunkt verbinden.

Um Kunden, die sich über VPC-Endpunkte mit Schnittstellen verbinden, ähnliche Funktionen zu bieten, füllt Amazon Keyspaces die `system.peers` Tabelle in Ihrem Konto mit einem Eintrag für jede Availability Zone, in der ein VPC-Endpunkt verfügbar ist. Um verfügbare Schnittstellen-VPC-Endpunkte in der `system.peers` Tabelle zu suchen und zu speichern, verlangt Amazon Keyspaces, dass Sie der IAM-Entität, die für die Verbindung mit Amazon Keyspaces verwendet wird, Zugriffsberechtigungen gewähren, um Ihre VPC nach den Endpunkt- und Netzwerkschnittstelleninformationen abzufragen.

### Important

Wenn Sie die `system.peers` Tabelle mit Ihren verfügbaren Schnittstellen-VPC-Endpunkten füllen, wird der Lastenausgleich verbessert und der Lese-/Schreibdurchsatz erhöht. Es wird für alle Clients empfohlen, die über VPC-Endpunkte auf Amazon Keyspaces zugreifen, und ist für Apache Spark erforderlich.

Um der IAM-Entität, die für die Verbindung mit Amazon Keyspaces verwendet wird, Berechtigungen zum Abrufen der erforderlichen VPC-Endpunktinformationen für die Schnittstelle zu gewähren, können Sie Ihre bestehende IAM-Rolle oder Benutzerrichtlinie aktualisieren oder eine neue IAM-Richtlinie erstellen, wie im folgenden Beispiel gezeigt.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ListVPCEndpoints",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeVpcEndpoints"
],
 "Resource": "*"
 }
]
}
```

### Note

Die verwalteten Richtlinien `AmazonKeyspacesReadOnlyAccess_v2` `AmazonKeyspacesFullAccess` enthalten die erforderlichen Berechtigungen, damit Amazon Keyspaces auf die Amazon EC2 EC2-Instance zugreifen kann, um Informationen über verfügbare Schnittstellen-VPC-Endpunkte zu lesen.

Um zu überprüfen, ob die Richtlinie korrekt eingerichtet wurde, fragen Sie die `system.peers` Tabelle nach Netzwerkinformationen ab. Wenn die `system.peers` Tabelle leer ist, könnte dies darauf hinweisen, dass die Richtlinie nicht erfolgreich konfiguriert wurde oder dass Sie das Kontingent für die Anforderungsrate `DescribeNetworkInterfaces` und `DescribeVPCEndpoints` API-Aktionen überschritten haben. `DescribeVPCEndpoints` fällt in die `Describe*` Kategorie und wird als nicht mutierende Aktion angesehen.

`DescribeNetworkInterfaces` fällt in die Untergruppe der ungefilterten und nicht paginierten, nicht mutierenden Aktionen, für die unterschiedliche Quoten gelten. Weitere Informationen finden Sie in der Amazon EC2 EC2-API-Referenz unter [Bucket-Größen und Nachfüllraten anfordern](#).

Wenn Sie eine leere Tabelle sehen, versuchen Sie es einige Minuten später erneut, um Probleme mit der Quote für Anfragen auszuschließen. Informationen dazu, ob Sie die VPC-Endpoints korrekt konfiguriert haben, finden Sie unter [the section called "Verbindungsfehler am VPC-Endpunkt"](#) Wenn Ihre Abfrage Ergebnisse aus der Tabelle zurückgibt, wurde Ihre Richtlinie korrekt konfiguriert.

## Steuerung des Zugriffs auf VPC-Schnittstellen-Endpunkte für Amazon Keyspaces

Mit VPC-Endpunktrichtlinien können Sie den Zugriff auf Ressourcen auf zwei Arten steuern:

- IAM-Richtlinie — Sie können die Anfragen, Benutzer oder Gruppen steuern, die über einen bestimmten VPC-Endpunkt auf Amazon Keyspaces zugreifen dürfen. Sie können dies tun, indem Sie einen [Bedingungsschlüssel](#) in der Richtlinie verwenden, der einem IAM-Benutzer, einer IAM-Gruppe oder einer IAM-Rolle zugeordnet ist.
- VPC-Richtlinie — Sie können kontrollieren, welche VPC-Endpunkte Zugriff auf Ihre Amazon Keyspaces-Ressourcen haben, indem Sie ihnen Richtlinien zuordnen. Um den Zugriff auf einen bestimmten Schlüsselraum oder eine Tabelle einzuschränken, sodass nur Datenverkehr über einen bestimmten VPC-Endpunkt zugelassen wird, bearbeiten Sie die bestehende IAM-Richtlinie, die den Ressourcenzugriff einschränkt, und fügen Sie diesen VPC-Endpunkt hinzu.

Im Folgenden finden Sie Beispiele für Endpunktrichtlinien für den Zugriff auf Amazon Keyspaces-Ressourcen.

- Beispiel für eine IAM-Richtlinie: Beschränken Sie den gesamten Zugriff auf eine bestimmte Amazon Keyspaces-Tabelle, sofern der Datenverkehr nicht vom angegebenen VPC-Endpunkt stammt — Diese Beispielrichtlinie kann einem IAM-Benutzer, einer IAM-Rolle oder einer IAM-Gruppe zugewiesen werden. Es schränkt den Zugriff auf eine bestimmte Amazon Keyspaces-Tabelle ein, es sei denn, der eingehende Verkehr stammt von einem bestimmten VPC-Endpunkt.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "UserOrRolePolicyToDenyAccess",
 "Action": "cassandra:*",
 "Effect": "Deny",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
],
 "Condition": { "StringNotEquals" : { "aws:sourceVpce": "vpce-abc123" } }
 }
]
}
```

**Note**

Um den Zugriff auf eine bestimmte Tabelle einzuschränken, müssen Sie auch den Zugriff auf die Systemtabellen einbeziehen. Systemtabellen sind schreibgeschützt.

- **Beispiel für eine VPC-Richtlinie: Schreibgeschützter Zugriff** — Diese Beispielrichtlinie kann an einen VPC-Endpunkt angehängt werden. (Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Amazon VPC-Ressourcen](#)). Es beschränkt Aktionen auf den schreibgeschützten Zugriff auf Amazon Keyspaces-Ressourcen über den VPC-Endpunkt, mit dem es verbunden ist.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ReadOnly",
 "Principal": "*",
 "Action": [
 "cassandra:Select"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
}
```

- **Beispiel für eine VPC-Richtlinie: Beschränken Sie den Zugriff auf eine bestimmte Amazon Keyspaces-Tabelle** — Diese Beispielrichtlinie kann an einen VPC-Endpunkt angehängt werden. Es schränkt den Zugriff auf eine bestimmte Tabelle über den VPC-Endpunkt ein, an den sie angehängt ist.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "RestrictAccessToTable",
 "Principal": "*",
 "Action": "cassandra:*",
 "Effect": "Allow",
 "Resource": [
```

```

 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
}
]
}

```

### Note

Um den Zugriff auf eine bestimmte Tabelle einzuschränken, müssen Sie auch den Zugriff auf die Systemtabellen einbeziehen. Systemtabellen sind schreibgeschützt.

## Verfügbarkeit

Amazon Keyspaces unterstützt die Verwendung von VPC-Endpunkten mit Schnittstellen überall dort, AWS-Regionen wo der Service verfügbar ist. Weitere Informationen finden Sie unter [???](#).

## VPC-Endpunktrichtlinien und Amazon point-in-time Keyspaces-Wiederherstellung (PITR)

Wenn Sie IAM-Richtlinien mit [Bedingungsschlüsseln](#) verwenden, um den eingehenden Datenverkehr einzuschränken, schlägt der Vorgang zur Wiederherstellung der Tabelle möglicherweise fehl. Wenn Sie beispielsweise den Quelldatenverkehr mithilfe von `aws:SourceVpce` Bedingungsschlüsseln auf bestimmte VPC-Endpunkte beschränken, schlägt die Tabellenwiederherstellung fehl. Damit Amazon Keyspaces im Namen Ihres Principals einen Wiederherstellungsvorgang durchführen kann, müssen Sie Ihrer IAM-Richtlinie einen `aws:ViaAWSService` Bedingungsschlüssel hinzufügen. Der `aws:ViaAWSService` Bedingungsschlüssel ermöglicht den Zugriff, wenn ein AWS Dienst unter Verwendung der Anmeldeinformationen des Prinzipals eine Anfrage stellt. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungsschlüssel](#) im IAM-Benutzerhandbuch. Die folgende Richtlinie ist ein Beispiel dafür.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CassandraAccessForVPCE",
 "Effect": "Allow",

```

```

 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "false"
 },
 "StringEquals": {
 "aws:SourceVpce": [
 "vpce-12345678901234567"
]
 }
 }
 },
 {
 "Sid": "CassandraAccessForAwsService",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "true"
 }
 }
 }
]
}

```

## Häufige Fehler und Warnungen

Wenn Sie Amazon Virtual Private Cloud verwenden und eine Verbindung zu Amazon Keyspaces herstellen, wird möglicherweise die folgende Warnung angezeigt.

```

Control node cassandra.us-east-1.amazonaws.com/1.111.111.111:9142 has an entry
for itself in system.peers: this entry will be ignored. This is likely due to a
misconfiguration;
please verify your rpc_address configuration in cassandra.yaml on all nodes in your
cluster.

```

Diese Warnung tritt auf, weil die `system.peers` Tabelle Einträge für alle Amazon VPC-Endpoints enthält, zu deren Anzeige Amazon Keyspaces berechtigt ist, einschließlich des Amazon VPC-Endpoints, über den Sie verbunden sind. Sie können diese Warnung getrost ignorieren.



Informationen zu anderen Fehlern finden Sie unter [the section called “Verbindungsfehler am VPC-Endpunkt”](#).

## Konfigurations- und Schwachstellenanalyse für Amazon Keyspaces

AWS übernimmt grundlegende Sicherheitsaufgaben wie Betriebssystemsicherheit und Datenbank-Patching, Firewall-Konfiguration und Notfallwiederherstellung. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Modell der [geteilten Verantwortung Modell](#) der der
- [Amazon Web Services: Übersicht über Sicherheitsverfahren](#) (Whitepaper)

## Best Practices für die Sicherheit in Amazon Keyspaces

Amazon Keyspaces (für Apache Cassandra) enthält eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

### Themen

- [Best Practices für die vorbeugende Sicherheitsmaßnahmen für Amazon Keyspaces](#)
- [Detective Practices für Amazon Keyspaces](#)

## Best Practices für die vorbeugende Sicherheitsmaßnahmen für Amazon Keyspaces

Die folgenden bewährten Methoden für die Sicherheit gelten als vorbeugend, da sie Ihnen helfen können, Sicherheitsvorfälle in Amazon Keyspaces zu antizipieren und zu verhindern.

### Verschlüsselung im Ruhezustand verwenden

Amazon Keyspaces verschlüsselt im Ruhezustand mithilfe von Verschlüsselungsschlüsseln, die in [AWS Key Management Service\(AWS KMS\)](#) gespeichert sind. Dies bietet eine zusätzliche Datenschutzebene, indem Ihre Daten vor unbefugtem Zugriff auf den zugrunde liegenden Speicher geschützt werden.

Standardmäßig verwendet Amazon Keyspaces einen AWS-eigenen Schlüssel zum Verschlüsseln aller Ihrer Tabellen. Wenn dieser Schlüssel nicht existiert, wird er für Sie erstellt. Die Standardschlüssel des -Service können nicht deaktiviert werden.

Alternativ können Sie einen vom [Kunden verwalteten Schlüssel](#) für die Verschlüsselung im Ruhezustand verwenden. Weitere Informationen finden Sie unter [Amazon Keyspaces](#)

#### Verwenden von IAM-Rollen zum Authentifizieren des Zugriffs auf Amazon Keyspaces

Damit Benutzer, Anwendungen und andere AWS -Services auf Amazon Keyspaces zugreifen können, müssen sie gültige AWS -Anmeldeinformationen in ihre AWS API-Anfragen aufnehmen. Sie sollten AWS-Anmeldeinformationen nicht direkt in der Anwendung oder einer EC2-Instance speichern. Dies sind langfristige Anmeldeinformationen, die nicht automatisch rotiert werden und daher erhebliche geschäftliche Auswirkungen haben können, wenn sie kompromittiert werden. Eine IAM-Rolle ermöglicht es Ihnen, temporäre Zugriffsschlüssel zu erhalten, mit denen Sie auf die AWS-Services und Ressourcen zugreifen können.

Weitere Informationen finden Sie unter [IAM-Rollen](#).

#### Verwenden von IAM-Richtlinien für die Amazon Keyspaces-Basisautorisierung

Beim Erteilen von Berechtigungen entscheiden Sie, wer die Berechtigungen erhält, für welche Amazon Keyspaces-APIs die Berechtigungen gelten und welche spezifischen Aktionen zu diesen Ressourcen gestattet werden sollen. Die Implementierung der geringsten Rechte ist der Schlüssel zur Verringerung der Sicherheitsrisiken und der Auswirkungen, die sich aus Fehlern oder böswilligen Absichten ergeben können.

Weisen Sie IAM-Identitäten (d. h. Benutzern, Gruppen und Rollen) Berechtigungsrichtlinien zu und erteilen Sie somit Berechtigungen zur Durchführung von Operationen für Amazon-Keyspaces-Ressourcen.

Sie können dies wie folgt tun:

- [AWS-verwaltete \(vordefinierte\) Richtlinien](#)
- [Kundenverwaltete Richtlinien](#)

#### Verwenden von IAM-Richtlinienbedingungen für eine differenzierte Zugriffssteuerung

Wenn Sie Berechtigungen in Amazon Keyspaces gewähren, können Sie Bedingungen angeben, die bestimmen, wie eine Berechtigungsrichtlinie wirksam wird. Die Implementierung der geringsten Rechte ist der Schlüssel zur Verringerung der Sicherheitsrisiken und der Auswirkungen, die sich aus Fehlern oder böswilligen Absichten ergeben können.

Sie können Bedingungen angeben, wenn Sie Berechtigungen mithilfe einer IAM-Richtlinie erteilen. Sie können z. B. Folgendes tun:

- Erteilen Sie Berechtigungen, um Benutzern schreibgeschützten Zugriff auf bestimmte Schlüsselbereiche oder Tabellen zu gewähren.
- Gewähren Sie einem Benutzer auf der Grundlage der Identität dieses Benutzers Schreibzugriff auf eine bestimmte Tabelle.

Weitere Informationen finden Sie unter [Identitätsbasierte -Richtlinien](#).

## Clientseitige Verschlüsselung in Betracht ziehen

Wenn Sie empfindliche oder vertrauliche Daten in Amazon Keyspaces speichern, sollten Sie diese Daten möglicherweise so nahe wie möglich an ihrem Ursprung verschlüsseln, damit Ihre Daten während des gesamten Lebenszyklus geschützt sind. Die Verschlüsselung Ihrer sensiblen Daten während der Übertragung und im Ruhezustand stellt sicher, dass Ihre Klartextdaten nicht für Dritte verfügbar sind.

## Detective Practices für Amazon Keyspaces

Die folgenden bewährten Methoden für die Sicherheit gelten als aufdeckend, da sie dabei helfen können, potenzielle Sicherheitsschwächen und Vorfälle zu erkennen.

Wird verwendet AWS CloudTrail, um die Verwendung von AWS Key Management Service (AWS KMS) AWS KMS -Schlüsseln zu überwachen

Wenn Sie einen vom [Kunden verwalteten AWS KMS Schlüssel](#) für die Verschlüsselung im Ruhezustand verwenden, ist die Verwendung dieses Schlüssels bei protokolliert AWS CloudTrail. CloudTrail bietet Einblick in die Benutzeraktivitäten, indem die in Ihrem Konto ausgeführten Aktionen aufgezeichnet werden. CloudTrail zeichnet wichtige Informationen zu jeder Aktion auf, einschließlich der Person, die Anforderung gestellt hat, der verwendeten Services, der ausgeführten Aktionen, der Parameter für die Aktionen und der vom AWS -Service zurückgegebenen Antwortelemente. Diese Informationen helfen Ihnen, Änderungen zu verfolgen, die an Ihren AWS -Ressourcen und Fehlerbehebung bei Betriebsproblemen durchgeführt wurden. CloudTrail erleichtert die Einhaltung interner Richtlinien und gesetzlicher Standards.

Sie können es verwenden CloudTrail , um die Schlüsselnutzung zu überprüfen. CloudTrail erstellt Protokolldateien, die einen Verlauf der AWS -API-Aufrufe und zugehörigen Ereignissen für Ihr Konto enthalten. Diese Protokolldateien enthalten alle AWS KMS -API-Anforderungen, die mit

der -Konsole, der AWS SDKs und Befehlszeilen-Tools gesendet wurden, sowie diejenigen, die über integrierte AWS -Services erfolgt sind. Sie können diese Protokolldateien verwenden, um Informationen über den Zeitpunkt der Verwendung des AWS KMS Schlüssels zu erhalten, sowie Informationen über die angeforderte Operation, die Identität des Auftraggebers und die IP-Adresse, von der aus die Anforderung gesendet wurde. Weitere Informationen finden Sie unter [Protokollierung von AWS Key Management Service-API-Aufrufen mit AWS CloudTrail](#) im [AWS CloudTrail-Benutzerhandbuch](#).

Wird verwendet CloudTrail, um die Operationen Keyspaces Definition Language (Data Definition Language) zu überwachen

CloudTrail bietet Einblick in die Benutzeraktivitäten, indem die in Ihrem Konto ausgeführten Aktionen aufgezeichnet werden. CloudTrail zeichnet wichtige Informationen zu jeder Aktion auf, einschließlich der Person, die Anforderung gestellt hat, der verwendeten Services, der ausgeführten Aktionen, der Parameter für die Aktionen und der vom AWS -Service zurückgegebenen Antwortelemente. Diese Informationen helfen Ihnen, Änderungen zu verfolgen, die an Ihren AWS -Ressourcen und Fehlerbehebung bei Betriebsproblemen durchgeführt wurden. CloudTrail erleichtert die Einhaltung interner Richtlinien und gesetzlicher Standards.

Alle Amazon Keyspaces [DDL-Operationen](#) werden CloudTrail automatisch angemeldet. Mit DDL-Verschlüsselung können Sie Amazon Keyspaces und Tabellen für Amazon Keyspaces erstellen und verwalten.

Wenn eine Aktivität in Amazon Keyspaces auftritt, wird diese Aktivität in einem CloudTrail - Ereignis zusammen mit anderen AWS Serviceereignissen im Ereignisverlauf protokolliert. Weitere Informationen finden Sie unter [Logging Amazon Keyspaces-Operationen mithilfe von AWS CloudTrail](#). Sie können die neusten Ereignisse in Ihr(em) AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Zur kontinuierlichen Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich der Ereignisse für Amazon Keyspaces, erstellen Sie einen [Trail](#). Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien in einem Amazon-Simple-Storage-Service Simple Storage Service (Amazon S3) Bucket. Wenn Sie einen Pfad auf der Konsole anlegen, gilt dieser standardmäßig für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien für den von Ihnen angegebenen S3 Bucket bereit. Darüber hinaus können Sie andere AWS -Services konfigurieren, um die in den CloudTrail - Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren.

## Markieren Sie Ihre Amazon Keyspaces-Ressourcen zur Identifizierung und Automatisierung

Sie können Metadaten Ihren AWS-Ressourcen in Form von Tags zuweisen. Jedes Tag ist eine einfache Bezeichnung, die aus einem benutzerdefinierten Schlüssel und einem optionalen Wert besteht, der das Verwalten, Suchen und Filtern von Ressourcen erleichtern kann.

Tagging ermöglicht die Implementierung gruppierter Steuerelemente. Obwohl es keine inhärenten Typen von Tags gibt, können Sie Ressourcen nach Zweck, Besitzer, Umgebung oder anderen Kriterien kategorisieren. Im Folgenden sind einige Beispiele aufgeführt:

- Zugriff — Wird verwendet, um den Zugriff auf Amazon Keyspaces-Ressourcen basierend auf Tags zu steuern. Weitere Informationen finden Sie unter [the section called “Autorisierung basierend auf Amazon Keyspaces-Tags”](#).
- Sicherheit — Wird verwendet, um Anforderungen wie Datenschutzeinstellungen zu bestimmen.
- Vertraulichkeit — Eine Kennung für die spezifische Datenvertraulichkeitsebene, die eine Ressource unterstützt
- Umgebung – Wird verwendet, um zwischen Entwicklungs-, Test- und Produktionsinfrastruktur zu unterscheiden.

Weitere Informationen finden Sie unter [AWSTagging-Strategien](#) und [Hinzufügen von Tags und Labels zu Ressourcen](#).

# CQL-Sprachreferenz für Amazon Keyspaces (für Apache Cassandra)

Nachdem Sie eine Verbindung zu einem Amazon Keyspaces-Endpunkt (für Apache Cassandra) hergestellt haben, verwenden Sie Cassandra Query Language (CQL), um mit Ihrer Datenbank zu arbeiten. CQL ähnelt in vielerlei Hinsicht der Structured Query Language (SQL).

## Themen

- [Elemente der Cassandra Query Language \(CQL\) in Amazon Keyspaces](#)
- [DDL-Anweisungen \(Datendefinitionssprache\) in Amazon Keyspaces](#)
- [DML-Anweisungen \(Datenmanipulationssprache\) in Amazon Keyspaces](#)
- [Integrierte Funktionen in Amazon Keyspaces](#)

## Elemente der Cassandra Query Language (CQL) in Amazon Keyspaces

Erfahren Sie mehr über die Cassandra Query Language (CQL) -Elemente, die von Amazon Keyspaces unterstützt werden, darunter Bezeichner, Konstanten, Begriffe und Datentypen.

## Themen

- [IDs](#)
- [Konstanten](#)
- [Bedingungen](#)
- [Datentypen](#)
- [JSON-Kodierung von Amazon Keyspaces-Datentypen](#)

## IDs

Identifikatoren (oder Namen) werden verwendet, um Tabellen, Spalten und andere Objekte zu identifizieren. Ein Bezeichner kann in Anführungszeichen gesetzt werden oder nicht. Folgendes gilt.

```
identifizier ::= unquoted_identifizier | quoted_identifizier
```

```
unquoted_identifizier ::= re('[a-zA-Z][a-zA-Z0-9]*')
quoted_identifizier ::= ''' (any character where " can appear if doubled)+ '''
```

## Konstanten

Die folgenden Konstanten sind definiert.

```
constant ::= string | integer | float | boolean | uuid | blob | NULL
string ::= '\' (any character where ' can appear if doubled)+ '\'
 '$$' (any character other than '$$') '$$'
integer ::= re('-?[0-9]+')
float ::= re('-?[0-9]+(\.[0-9]*)?([eE][+-]?[0-9+]?)') | NAN | INFINITY
boolean ::= TRUE | FALSE
uuid ::= hex{8}-hex{4}-hex{4}-hex{4}-hex{12}
hex ::= re("[0-9a-fA-F]")
blob ::= '0' ('x' | 'X') hex+
```

## Bedingungen

Ein Begriff bezeichnet die Art von Werten, die unterstützt werden. Begriffe werden wie folgt definiert.

```
term ::= constant | literal | function_call | arithmetic_operation |
 type_hint | bind_marker
literal ::= collection_literal | tuple_literal
function_call ::= identifizier '(' [term (',' term)*] ')
arithmetic_operation ::= '-' term | term ('+' | '-' | '*' | '/' | '%') term
```

## Datentypen

Amazon Keyspaces unterstützt die folgenden Datentypen:

### Zeichenfolgen-Typen

| Datentyp           | Beschreibung                        |
|--------------------|-------------------------------------|
| <code>ascii</code> | Stellt eine ASCII-Zeichenfolge dar. |

| Datentyp             | Beschreibung                                                                               |
|----------------------|--------------------------------------------------------------------------------------------|
| <code>text</code>    | Stellt eine UTF-8-kodierte Zeichenfolge dar.                                               |
| <code>varchar</code> | Stellt eine UTF-8-kodierte Zeichenfolge dar (ist ein Alias für <code>varchar.text</code> ) |

## Numerische Typen

| Datentyp             | Beschreibung                                                                                                                                |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>bigint</code>  | Stellt einen 64-Bit-Wert vom Typ Long mit Vorzeichen dar.                                                                                   |
| <code>counter</code> | Stellt einen 64-Bit-Integer-Zähler mit Vorzeichen dar. Weitere Informationen finden Sie unter <a href="#">the section called “Zähler”</a> . |
| <code>decimal</code> | Stellt eine Dezimalzahl mit variabler Genauigkeit dar.                                                                                      |
| <code>double</code>  | Stellt einen 64-Bit-IEEE-754-Gleitkommawert dar.                                                                                            |
| <code>float</code>   | Stellt einen 32-Bit-Gleitkommawert nach IEEE 754 dar.                                                                                       |
| <code>int</code>     | Stellt einen 32-Bit-Int-Wert mit Vorzeichen dar.                                                                                            |
| <code>varint</code>  | Stellt eine Ganzzahl mit beliebiger Genauigkeit dar.                                                                                        |

## Zähler

Eine `counter` Spalte enthält eine 64-Bit-Ganzzahl mit Vorzeichen. Der Zählerwert wird mithilfe der [the section called “UPDATE”](#) Anweisung inkrementiert oder dekrementiert und kann nicht direkt festgelegt werden. Das macht `counter` Spalten nützlich, um Zählungen nachzuverfolgen. Mithilfe von Zählern können Sie beispielsweise die Anzahl der Einträge in einer Protokolldatei oder die



Anzahl der Aufrufe eines Beitrags in einem sozialen Netzwerk verfolgen. Für `counter` Spalten gelten die folgenden Einschränkungen:

- Eine Spalte vom Typ `counter` kann nicht Teil `primary key` einer Tabelle sein.
- In einer Tabelle, die eine oder mehrere Spalten des Typs `counter` enthält, müssen alle Spalten in dieser Tabelle vom Typ `counter` sein.

In Fällen, in denen eine Zähleraktualisierung fehlschlägt (z. B. aufgrund von Timeouts oder Verbindungsverlust mit Amazon Keyspaces), weiß der Client nicht, ob der Zählerwert aktualisiert wurde. Wenn die Aktualisierung erneut versucht wird, wird die Aktualisierung des Zählerwerts möglicherweise ein zweites Mal angewendet.

## Blob-Typ

| Datentyp          | Beschreibung               |
|-------------------|----------------------------|
| <code>blob</code> | Stellt beliebige Byte dar. |

## Typ BOOLEAN

| Datentyp             | Beschreibung                                          |
|----------------------|-------------------------------------------------------|
| <code>boolean</code> | Steht für <code>true</code> oder <code>false</code> . |

## Zeitbezogene Typen

| Datentyp               | Beschreibung                                                                                                                    |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <code>timestamp</code> | 64-Bit-Ganzzahl mit Vorzeichen, die Datum und Uhrzeit seit der Epoche (1. Januar 1970 um 00:00:00 GMT) in Millisekunden angibt. |
| <code>timeuuid</code>  | <a href="#">Stellt eine UUID der Version 1 dar.</a>                                                                             |

## Sammlungstypen

| Datentyp          | Beschreibung                                                                      |
|-------------------|-----------------------------------------------------------------------------------|
| <code>list</code> | Stellt eine geordnete Sammlung literaler Elemente dar.                            |
| <code>map</code>  | Stellt eine ungeordnete Sammlung von Schlüssel-Wert-Paaren dar.                   |
| <code>set</code>  | Stellt eine ungeordnete Sammlung von einem oder mehreren literalen Elementen dar. |

Sie deklarieren eine Auflistungsspalte, indem Sie den Sammlungstyp verwenden, gefolgt von einem anderen Datentyp (z. B. `TEXT` oder `INT`) in spitzen Klammern. Sie können eine Spalte mit einem `SET of` erstellen `TEXT`, oder Sie können Paare aus `MAP of` `TEXT` und `INT` Schlüssel-Wert-Paaren erstellen, wie im folgenden Beispiel gezeigt.

```
SET <TEXT>
MAP <TEXT, INT>
```

Eine nicht eingefrorene Sammlung ermöglicht es Ihnen, Aktualisierungen an jedem einzelnen Sammlungselement vorzunehmen. Clientseitige Zeitstempel und Time-to-Live-Einstellungen (TTL) werden für einzelne Elemente gespeichert.

Wenn Sie das `FROZEN` Schlüsselwort für einen Sammlungstyp verwenden, werden die Werte der Sammlung zu einem einzigen unveränderlichen Wert serialisiert, und Amazon Keyspaces behandelt sie wie einen `BLOB`. Dies ist eine eingefrorene Sammlung. Eine `INSERT UPDATE` Oder-Anweisung überschreibt die gesamte eingefrorene Sammlung. Sie können einzelne Elemente in einer eingefrorenen Sammlung nicht aktualisieren.

Clientseitige Zeitstempel und TTL-Einstellungen (Time to Live) gelten für die gesamte eingefrorene Sammlung, nicht für einzelne Elemente. `FROZEN` Sammlungsspalten können Teil einer Tabelle sein. `PRIMARY KEY`

Sie können eingefrorene Sammlungen verschachteln. Sie können beispielsweise `MAP` innerhalb von `a` definieren, `SET` wenn der `MAP` das `FROZEN` Schlüsselwort verwendet, wie im folgenden Beispiel gezeigt.

```
SET <FROZEN> <MAP <TEXT, INT>>>
```

Amazon Keyspaces unterstützt standardmäßig die Verschachtelung von bis zu fünf Ebenen eingefrorener Sammlungen. Weitere Informationen finden Sie unter [the section called “Service Quotas für Amazon Keyspaces”](#). Weitere Informationen zu den funktionalen Unterschieden mit Apache Cassandra finden Sie unter [the section called “FROZENSammlungen”](#). Weitere Hinweise zur CQL-Syntax finden Sie unter [the section called “CREATE TABLE”](#) und [the section called “ALTER TABLE”](#).

## Typ „Tupel“

Der `tuple` Datentyp steht für eine begrenzte Gruppe literaler Elemente. Sie können ein Tupel als Alternative zu einem verwenden. `user defined type` Sie müssen das `FROZEN` Schlüsselwort nicht für Tupel verwenden. Das liegt daran, dass ein Tupel immer eingefroren ist und Sie Elemente nicht einzeln aktualisieren können.

## Andere Typen

| Datentyp          | Beschreibung                                                                           |
|-------------------|----------------------------------------------------------------------------------------|
| <code>inet</code> | Eine Zeichenfolge, die eine IP-Adresse darstellt , entweder im IPv4- oder IPv6-Format. |

## Statisch

In einer Amazon Keyspaces-Tabelle mit Clusterspalten können Sie das `STATIC` Schlüsselwort verwenden, um eine statische Spalte beliebigen Typs zu erstellen.

Die folgende Aussage ist ein Beispiel dafür.

```
my_column INT STATIC
```

Weitere Hinweise zum Arbeiten mit statischen Spalten finden Sie unter [the section called “Statische Spalten”](#).

## JSON-Kodierung von Amazon Keyspaces-Datentypen

Amazon Keyspaces bietet dieselben JSON-Datentypzuordnungen wie Apache Cassandra. In der folgenden Tabelle werden die Datentypen beschrieben, die Amazon Keyspaces in `INSERT JSON` Anweisungen akzeptiert, und die Datentypen, die Amazon Keyspaces bei der Rückgabe von Daten mit der `SELECT JSON` Anweisung verwendet.

Für Einzelfeld-Datentypen wie `float`, `int` `UUID` `date`, und können Sie Daten auch als einfügen. `string` Für zusammengesetzte Datentypen und Sammlungen wie `tuple` `map` `list`, und können Sie Daten auch als JSON oder codiert `JSON string` einfügen.

| JSON-Datentyp        | In <b>INSERT JSON</b> Anweisungen akzeptierte Datentypen | In <b>SELECT JSON</b> Anweisungen zurückgegebene Datentypen | Hinweise                                                                                                              |
|----------------------|----------------------------------------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>ascii</code>   | <code>string</code>                                      | <code>string</code>                                         | Verwendet JSON-Zeichen-Escape-Zeichen <code>\u</code> .                                                               |
| <code>bigint</code>  | <code>integer</code> , <code>string</code>               | <code>integer</code>                                        | Die Zeichenfolge muss eine gültige 64-Bit-Ganzzahl sein.                                                              |
| <code>blob</code>    | <code>string</code>                                      | <code>string</code>                                         | Die Zeichenfolge sollte mit <code>beginnen</code> , <code>0x</code> gefolgt von einer geraden Anzahl von Hex-Ziffern. |
| <code>boolean</code> | <code>boolean</code> , <code>string</code>               | <code>boolean</code>                                        | Die Zeichenfolge muss entweder <code>true</code> oder <code>seinfalse</code> .                                        |
| <code>date</code>    | <code>string</code>                                      | <code>string</code>                                         | Datum im Format <code>YYYY-MM-DD</code> , Zeitzone UTC.                                                               |

| JSON-Datentyp        | In <b>INSERT JSON</b> Anweisungen akzeptierte Datentypen | In <b>SELECT JSON</b> Anweisungen zurückgegebene Datentypen | Hinweise                                                                                             |
|----------------------|----------------------------------------------------------|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <code>decimal</code> | <code>integer, float, string</code>                      | <code>float</code>                                          | Kann im clientseitigen Decoder die 32-Bit- oder 64-Bit-IEEE-754-Gleitkommagenauigkeit überschreiten. |
| <code>double</code>  | <code>integer, float, string</code>                      | <code>float</code>                                          | Die Zeichenfolge muss eine gültige Ganzzahl oder eine Fließkommazahl sein.                           |
| <code>float</code>   | <code>integer, float, string</code>                      | <code>float</code>                                          | Die Zeichenfolge muss eine gültige Ganzzahl oder eine Fließkommazahl sein.                           |
| <code>inet</code>    | <code>string</code>                                      | <code>string</code>                                         | IPv4- oder IPv6-Adresse.                                                                             |
| <code>int</code>     | <code>integer, string</code>                             | <code>integer</code>                                        | Die Zeichenfolge muss eine gültige 32-Bit-Ganzzahl sein.                                             |
| <code>list</code>    | <code>list, string</code>                                | <code>list</code>                                           | Verwendet die native JSON-Listendarstellung.                                                         |
| <code>map</code>     | <code>map, string</code>                                 | <code>map</code>                                            | Verwendet die native JSON-Kartendarstellung.                                                         |

| JSON-Datentyp          | In <b>INSERT JSON</b> Anweisungen akzeptierte Datentypen | In <b>SELECT JSON</b> Anweisungen zurückgegebene Datentypen | Hinweise                                                                                                                                                                                       |
|------------------------|----------------------------------------------------------|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>smallint</code>  | <code>integer, string</code>                             | <code>integer</code>                                        | Die Zeichenfolge muss eine gültige 16-Bit-Ganzzahl sein.                                                                                                                                       |
| <code>set</code>       | <code>list, string</code>                                | <code>list</code>                                           | Verwendet die native JSON-Listendarstellung.                                                                                                                                                   |
| <code>text</code>      | <code>string</code>                                      | <code>string</code>                                         | Verwendet JSON-Zeichen-Escape <u>\</u> .                                                                                                                                                       |
| <code>time</code>      | <code>string</code>                                      | <code>string</code>                                         | Tageszeit im Format <code>HH-MM-SS[.ffffff]</code> .                                                                                                                                           |
| <code>timestamp</code> | <code>integer, string</code>                             | <code>string</code>                                         | Ein Zeitstempel. Zeichenkettenkonstanten ermöglichen es Ihnen, Zeitstempel als Datumsangaben zu speichern. Datumsstempel mit Format <code>YYYY-MM-DD HH:MM:SS.SSS</code> werden zurückgegeben. |
| <code>timeuuid</code>  | <code>string</code>                                      | <code>string</code>                                         | Geben Sie 1 UUID ein. Informationen <a href="#">constants</a> zum UUID-Format finden Sie unter.                                                                                                |

| JSON-Datentyp        | In <b>INSERT JSON</b> Anweisungen akzeptierte Datentypen | In <b>SELECT JSON</b> Anweisungen zurückgegebene Datentypen | Hinweise                                                                                  |
|----------------------|----------------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <code>tinyint</code> | <code>integer, string</code>                             | <code>integer</code>                                        | Die Zeichenfolge muss eine gültige 8-Bit-Ganzzahl sein.                                   |
| <code>tuple</code>   | <code>list, string</code>                                | <code>list</code>                                           | Verwendet die native JSON-Listendarstellung.                                              |
| <code>uuid</code>    | <code>string</code>                                      | <code>string</code>                                         | Informationen <a href="#">constants</a> zum UUID-Format finden Sie unter.                 |
| <code>varchar</code> | <code>string</code>                                      | <code>string</code>                                         | Verwendet JSON-Zeichen-Escape. <code>\u</code>                                            |
| <code>varint</code>  | <code>integer, string</code>                             | <code>integer</code>                                        | Variable Länge; kann 32-Bit- oder 64-Bit-Ganzzahlen im clientseitigen Decoder überlaufen. |

## DDL-Anweisungen (Datendefinitionssprache) in Amazon Keyspaces

Data Definition Language (DDL) ist der Satz von Cassandra Query Language (CQL) -Anweisungen, die Sie verwenden, um Datenstrukturen in Amazon Keyspaces (für Apache Cassandra) zu verwalten, z. B. Schlüsselräume und Tabellen. Sie verwenden DDL, um diese Datenstrukturen zu erstellen, sie nach ihrer Erstellung zu ändern und sie zu entfernen, wenn sie nicht mehr verwendet werden. Amazon Keyspaces führt DDL-Operationen asynchron aus. Weitere Informationen darüber, wie Sie

überprüfen können, ob ein asynchroner Vorgang abgeschlossen wurde, finden Sie unter [the section called “Asynchrones Erstellen und Löschen von Schlüsselräumen und Tabellen”](#)

Die folgenden DDL-Anweisungen werden unterstützt:

- [SCHLÜSSELRAUM ERSTELLEN](#)
- [SCHLÜSSELRAUM ÄNDERN](#)
- [KEYSPACE LÖSCHEN](#)
- [TABELLE ERSTELLEN](#)
- [ALTER TABLE](#)
- [TABELLE WIEDERHERSTELLEN](#)
- [TABELLE LÖSCHEN](#)

Themen

- [Keyspaces](#)
- [Tabellen](#)

## Keyspaces

Ein Schlüsselraum gruppiert verwandte Tabellen, die für eine oder mehrere Anwendungen relevant sind. In Bezug auf ein relationales Datenbankmanagementsystem (RDBMS) ähneln Schlüsselräume in etwa Datenbanken, Tablespace oder ähnlichen Konstrukten.

### Note

In Apache Cassandra bestimmen Schlüsselräume, wie Daten zwischen mehreren Speicherknoten repliziert werden. Amazon Keyspaces ist jedoch ein vollständig verwalteter Service: Die Details der Speicherebene werden in Ihrem Namen verwaltet. Aus diesem Grund sind Schlüsselräume in Amazon Keyspaces nur logische Konstrukte und haben nichts mit dem zugrunde liegenden physischen Speicher zu tun.

Informationen zu Kontingentbeschränkungen und Einschränkungen für Amazon Keyspaces-Keyspaces finden Sie unter [Kontingente](#)

Anweisungen für Schlüsselräume



- [SCHLÜSSELRAUM ERSTELLEN](#)
- [SCHLÜSSELRAUM ÄNDERN](#)
- [SCHLÜSSELRAUM LÖSCHEN](#)

## SCHLÜSSELRAUM ERSTELLEN

Verwenden Sie die `CREATE KEYSPACE` Anweisung, um einen neuen Schlüsselraum zu erstellen.

### Syntax

```
create_keyspace_statement ::=
 CREATE KEYSPACE [IF NOT EXISTS] keyspace_name
 WITH options
```

Wobei gilt:

- *keyspace\_name* ist der Name des zu erstellenden Schlüsselraums.
- Optionen sind eine oder mehrere der folgenden Optionen:
  - `REPLICATION`— Eine Map, die die Replikationsstrategie für den Schlüsselraum angibt:
    - `SingleRegionStrategy`— Für einen Schlüsselraum mit nur einer Region. (Erforderlich)
    - `NetworkTopologyStrategy`— Geben Sie mindestens zwei und bis zu sechs an. AWS-Regionen Der Replikationsfaktor für jede Region ist drei. (Optional)
  - `DURABLE_WRITES`— Schreibvorgänge in Amazon Keyspaces sind immer dauerhaft, sodass diese Option nicht erforderlich ist. Falls angegeben, muss der Wert jedoch sein. `true`
  - `TAGS`— Eine Liste von Schlüssel-Wert-Paar-Tags, die bei der Erstellung an die Ressource angehängt werden sollen. (Optional)

### Beispiel

Erstellen Sie einen Schlüsselraum wie folgt.

```
CREATE KEYSPACE my_keyspace
 WITH REPLICATION = {'class': 'SingleRegionStrategy'} and TAGS ={'key1':'val1',
 'key2':'val2'} ;
```

Um einen Schlüsselraum für mehrere Regionen zu erstellen, geben Sie mindestens zwei `NetworkTopologyStrategy` und bis zu sechs an und schließen Sie diese ein. AWS-Regionen Der Replikationsfaktor für jede Region ist drei.

```
CREATE KEYSPACE my_keyspace
 WITH REPLICATION = {'class':'NetworkTopologyStrategy', 'us-east-1':'3', 'ap-
southeast-1':'3', 'eu-west-1':'3'};
```

## SCHLÜSSELRAUM ÄNDERN

Verwenden Sie den `ALTER KEYSPACE`, um Tags zu einem Schlüsselraum hinzuzufügen oder daraus zu entfernen.

### Syntax

```
alter_keyspace_statement ::=
 ALTER KEYSPACE keyspace_name
 [[ADD | DROP] TAGS
```

Wobei gilt:

- *keyspace\_name* ist der Name des Schlüsselraums, der geändert werden soll.
- TAGS— Eine Liste von Schlüssel-Wert-Paar-Tags, die dem Schlüsselraum hinzugefügt oder daraus entfernt werden sollen.

### Beispiel

Ändern Sie einen Schlüsselraum wie folgt.

```
ALTER KEYSPACE "myGSGKeyspace" ADD TAGS {'key1':'val1', 'key2':'val2'};
```

## SCHLÜSSELRAUM LÖSCHEN

Verwenden Sie die `DROP KEYSPACE` Anweisung, um einen Schlüsselraum einschließlich seines gesamten Inhalts, z. B. Tabellen, zu entfernen.

### Syntax

```
drop_keyspace_statement ::=
 DROP KEYSPACE [IF EXISTS] keyspace_name
```

Wobei gilt:

- `keyspace_name` ist der Name des Schlüsselraums, der gelöscht werden soll.

Beispiel

```
DROP KEYSPACE "myGSGKeyspace";
```

## Tabellen

Tabellen sind die primären Datenstrukturen in Amazon Keyspaces. Daten in einer Tabelle sind in Zeilen und Spalten organisiert. Eine Teilmenge dieser Spalten wird verwendet, um die Partitionierung (und letztlich die Datenplatzierung) durch die Angabe eines Partitionsschlüssels zu bestimmen.

Eine weitere Gruppe von Spalten kann in Clusterspalten definiert werden, was bedeutet, dass sie als Prädikate an der Abfrageausführung teilnehmen können.

Standardmäßig werden neue Tabellen mit On-Demand-Durchsatzkapazität erstellt. Sie können den Kapazitätsmodus für neue und bestehende Tabellen ändern. Weitere Informationen zu den Durchsatzmodi mit Lese-/Schreibkapazität finden Sie unter [the section called “Lese-/Schreibkapazitätsmodi”](#)

Für Tabellen im Bereitstellungsmodus können Sie sie optional konfigurieren.

`AUTOSCALING_SETTINGS` Weitere Informationen zur auto Skalierung von Amazon Keyspaces und den verfügbaren Optionen finden Sie unter [the section called “Verwenden von CQL”](#).

Informationen zu Kontingentbeschränkungen und Einschränkungen für Amazon Keyspaces-Tabellen finden Sie unter [Kontingente](#).

Aussagen für Tabellen

- [CREATE TABLE](#)
- [ALTER TABLE](#)
- [TABELLE WIEDERHERSTELLEN](#)
- [DROP TABLE](#)

## CREATE TABLE

Verwenden Sie die `CREATE TABLE` Anweisung, um eine neue Tabelle zu erstellen.

## Syntax

```

create_table_statement ::= CREATE TABLE [IF NOT EXISTS] table_name
 '('
 column_definition
 (',' column_definition)*
 [',' PRIMARY KEY '(' primary_key ')']
 ')' [WITH table_options]

column_definition ::= column_name cql_type [FROZEN][STATIC][PRIMARY KEY]

primary_key ::= partition_key [',' clustering_columns]

partition_key ::= column_name
 | '(' column_name (',' column_name)* ')'

clustering_columns ::= column_name (',' column_name)*

table_options ::= [table_options]
 | CLUSTERING ORDER BY '(' clustering_order
 ')' [AND table_options]
 | options
 | CUSTOM_PROPERTIES
 | AUTOSCALING_SETTINGS
 | default_time_to_live
 | TAGS

clustering_order ::= column_name (ASC | DESC) (',' column_name (ASC | DESC))*

```

Wobei gilt:

- *table\_name* ist der Name der Tabelle, die erstellt werden soll.
- *column\_definition* besteht aus den folgenden Komponenten:
  - *column\_name*— Der Name der Spalte.
  - *cql\_type*— Ein Amazon Keyspaces-Datentyp (siehe [Datentypen](#)).
  - *FROZEN*— Kennzeichnet diese Spalte vom Typ `collection` (zum Beispiel, `LISTSET`, oder `MAP`) als eingefroren. Eine eingefrorene Sammlung wird in einen einzigen unveränderlichen Wert serialisiert und wie eine behandelt. BLOB Weitere Informationen finden Sie unter [the section called “Sammlungstypen”](#).


- *STATIC*— Kennzeichnet diese Spalte als statisch. Statische Spalten speichern Werte, die von allen Zeilen in derselben Partition gemeinsam genutzt werden.
- *PRIMARY KEY*— Bezeichnet diese Spalte als Primärschlüssel der Tabelle.
- *primary\_key* besteht aus den folgenden Komponenten:
  - *partition\_key*
  - *clustering\_columns*
- *partition\_key*:
  - Der Partitionsschlüssel kann eine einzelne Spalte oder ein zusammengesetzter Wert sein, der aus zwei oder mehr Spalten besteht. Der Partitionsschlüsselteil des Primärschlüssels ist erforderlich und bestimmt, wie Amazon Keyspaces Ihre Daten speichert.
- *clustering\_columns*:
  - Der optionale Teil der Clustering-Spalte Ihres Primärschlüssels bestimmt, wie die Daten innerhalb der einzelnen Partitionen gruppiert und sortiert werden.
- *table\_options* bestehen aus den folgenden Komponenten:
  - *CLUSTERING ORDER BY*— Die standardmäßige CLUSTERING-REIHENFOLGE für eine Tabelle setzt sich aus Ihren Gruppierungsschlüsseln in ASC (aufsteigender) Sortierrichtung zusammen. Geben Sie es an, um das Standard-Sortierverhalten zu überschreiben.
  - *CUSTOM\_PROPERTIES*— Eine Karte mit Einstellungen, die für Amazon Keyspaces spezifisch sind.
    - *capacity\_mode*: Gibt den Lese-/Schreibdurchsatzkapazitätsmodus für die Tabelle an. Die Optionen sind `throughput_mode:PAY_PER_REQUEST` und `throughput_mode:PROVISIONED`. Der bereitgestellte Kapazitätsmodus erfordert `read_capacity_units` und `write_capacity_units` als Eingaben. Der Standardwert ist `throughput_mode:PAY_PER_REQUEST`.
    - *client\_side\_timestamps*: Gibt an, ob clientseitige Zeitstempel für die Tabelle aktiviert oder deaktiviert sind. Die Optionen sind `{'status': 'enabled'}` und `{'status': 'disabled'}`. Wenn es nicht angegeben ist, ist die Standardeinstellung `status:disabled`. Nachdem clientseitige Zeitstempel für eine Tabelle aktiviert wurden, kann diese Einstellung nicht deaktiviert werden.
    - *encryption\_specification*: Gibt die Verschlüsselungsoptionen für die Verschlüsselung im Ruhezustand an. Wenn es nicht angegeben ist, ist die Standardeinstellung `encryption_type:AWS_OWNED_KMS_KEY`. Die vom Kunden verwaltete

Verschlüsselungsoption erfordert den AWS KMS Schlüssel im Format Amazon Resource Name (ARN) als Eingabe: `kms_key_identifizier:ARN:kms_key_identifizier:ARN`.

- `point_in_time_recovery`: Gibt an, ob die point-in-time Wiederherstellung für die Tabelle aktiviert oder deaktiviert ist. Die Optionen sind `status:enabled` und `status:disabled`. Wenn es nicht angegeben ist, ist die Standardeinstellung `status:disabled`.
- `replica_updates`: Gibt die Einstellungen einer Tabelle mit mehreren Regionen an, die spezifisch für eine AWS-Region sind. Bei einer Tabelle mit mehreren Regionen können Sie die Lesekapazität der Tabelle je nach Region unterschiedlich konfigurieren. AWS-Region Sie können dies tun, indem Sie die folgenden Parameter konfigurieren. Weitere Informationen und Beispiele finden Sie unter [the section called “Erstellen einer multiregionalen Tabelle mit bereitgestelltem Kapazitätsmodus und Auto Scaling \(CQL\)”](#).
- `region`— Das AWS-Region des Tabellenreplikats mit den folgenden Einstellungen:
  - `read_capacity_units`
- `TTL`: Aktiviert die benutzerdefinierten Time to Live-Einstellungen für die Tabelle. Verwenden Sie zum Aktivieren `status:enabled`. Der Standardwert ist `status:disabled`. Nachdem `TTL` es aktiviert wurde, können Sie es für die Tabelle nicht mehr deaktivieren.
- `AUTOSCALING_SETTINGS` beinhaltet die folgenden optionalen Einstellungen für Tabellen im Bereitstellungsmodus. Weitere Informationen und Beispiele finden Sie unter [the section called “Erstellen einer neuen Tabelle mit automatischer Skalierung mithilfe von CQL”](#).
- `provisioned_write_capacity_autoscaling_update`:
  - `autoscaling_disabled`— Um die auto Skalierung für die Schreibkapazität zu aktivieren, setzen Sie den Wert auf `false`. Der Standardwert ist `true`. (Optional)
  - `minimum_units`— Der Mindestwert für den Schreibdurchsatz, den die Tabelle immer unterstützen sollte. Der Wert muss zwischen 1 und dem maximalen Kontingent für den Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
  - `maximum_units`— Der maximale Schreibdurchsatz, den die Tabelle immer unterstützen sollte. Der Wert muss zwischen 1 und dem maximalen Kontingent für den Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
  - `scaling_policy`— Amazon Keyspaces unterstützt die Richtlinie zur Zielverfolgung. Das Auto Scaling-Ziel ist die bereitgestellte Schreibkapazität der Tabelle.
    - `target_tracking_scaling_policy_configuration`— Um die Zielverfolgungsrichtlinie zu definieren, müssen Sie den Zielwert definieren. Weitere Informationen zu Target Tracking und Cooldown-Perioden finden Sie unter [Target Tracking Scaling Policies](#) im Application Auto Scaling User Guide.

- `target_value`— Die Zielauslastungsrate der Tabelle. Die auto Skalierung von Amazon Keyspaces stellt sicher, dass das Verhältnis von verbrauchter Kapazität zu bereitgestellter Kapazität auf oder nahe diesem Wert bleibt. Sie definieren `target_value` als Prozentsatz. Ein Doppelter zwischen 20 und 90. (Erforderlich)
- `scale_in_cooldown`— Eine Abklingzeit in Sekunden zwischen den Skalierungsaktivitäten, sodass sich die Tabelle stabilisiert, bevor eine weitere Aktivitätsskala beginnt. Wenn kein Wert angegeben wird, ist der Standardwert 0. (Optional)
- `scale_out_cooldown`— Eine Abklingzeit in Sekunden zwischen Skalierungsaktivitäten, durch die sich die Tabelle stabilisiert, bevor eine weitere Scale-Out-Aktivität beginnt. Wenn kein Wert angegeben wird, ist der Standardwert 0. (Optional)
- `disable_scale_in`: Aboolean, das angibt, ob `scale-in` es für die Tabelle deaktiviert oder aktiviert ist. Dieser Parameter ist standardmäßig deaktiviert. Stellen Sie zum `scale-in` Einschalten den boolean Wert auf ein `FALSE`. Das bedeutet, dass die Kapazität für eine Tabelle in Ihrem Namen automatisch herunterskaliert wird. (Optional)
- `provisioned_read_capacity_autoscaling_update`:
  - `autoscaling_disabled`— Um die auto Skalierung für die Lesekapazität zu aktivieren, setzen Sie den Wert auf `false`. Der Standardwert ist `true`. (Optional)
  - `minimum_units`— Der Mindestdurchsatz, den die Tabelle immer unterstützen sollte. Der Wert muss zwischen 1 und dem maximalen Kontingent für den Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
  - `maximum_units`— Der maximale Durchsatz, den die Tabelle immer unterstützen sollte. Der Wert muss zwischen 1 und dem maximalen Kontingent für den Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
  - `scaling_policy`— Amazon Keyspaces unterstützt die Richtlinie zur Zielverfolgung. Das Auto Scaling-Ziel ist die bereitgestellte Lesekapazität der Tabelle.
  - `target_tracking_scaling_policy_configuration`— Um die Zielverfolgungsrichtlinie zu definieren, müssen Sie den Zielwert definieren. Weitere Informationen zu Target Tracking und Cooldown-Perioden finden Sie unter [Target Tracking Scaling Policies](#) im Application Auto Scaling User Guide.
  - `target_value`— Die Zielauslastungsrate der Tabelle. Die auto Skalierung von Amazon Keyspaces stellt sicher, dass das Verhältnis von verbrauchter Kapazität

zu bereitgestellter Kapazität auf oder nahe diesem Wert bleibt. Sie definieren `target_value` als Prozentsatz. Ein Doppelter zwischen 20 und 90. (Erforderlich)

- `scale_in_cooldown`— Eine Abklingzeit in Sekunden zwischen den Skalierungsaktivitäten, sodass sich die Tabelle stabilisiert, bevor eine weitere Aktivitätsskala beginnt. Wenn kein Wert angegeben wird, ist der Standardwert 0. (Optional)
  - `scale_out_cooldown`— Eine Abklingzeit in Sekunden zwischen Skalierungsaktivitäten, durch die sich die Tabelle stabilisiert, bevor eine weitere Scale-Out-Aktivität beginnt. Wenn kein Wert angegeben wird, ist der Standardwert 0. (Optional)
  - `disable_scale_in`: Aboolean, das angibt, ob `scale-in` es für die Tabelle deaktiviert oder aktiviert ist. Dieser Parameter ist standardmäßig deaktiviert. Stellen Sie zum `scale-in` Einschalten den boolean Wert auf ein `FALSE`. Das bedeutet, dass die Kapazität für eine Tabelle in Ihrem Namen automatisch herunterskaliert wird. (Optional)
  - `replica_updates`: Gibt die AWS-Region spezifischen Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen an. Bei einer Tabelle mit mehreren Regionen können Sie die Lesekapazität der Tabelle je nach Region unterschiedlich konfigurieren. AWS-Region Sie können dies tun, indem Sie die folgenden Parameter konfigurieren. Weitere Informationen und Beispiele finden Sie unter [the section called “Erstellen einer multiregionalen Tabelle mit bereitgestelltem Kapazitätsmodus und Auto Scaling \(CQL\)”](#).
  - `region`— Das AWS-Region des Tabellenreplikats mit den folgenden Einstellungen:
    - `provisioned_read_capacity_autoscaling_update`
    - `autoscaling_disabled`— Um die auto Skalierung für die Lesekapazität der Tabelle zu aktivieren, setzen Sie den Wert auf `false`. Der Standardwert ist `true`. (Optional)
-  **Note**

Auto Scaling für eine Tabelle mit mehreren Regionen muss für alle Replikate der Tabelle entweder aktiviert oder deaktiviert sein.
- `minimum_units`— Der Mindestwert an Lesedurchsatz, den die Tabelle immer unterstützen sollte. Der Wert muss zwischen 1 und dem maximalen Kontingent für den Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).



- `maximum_units`— Der maximale Lesedurchsatz, den die Tabelle immer unterstützen sollte. Der Wert muss zwischen 1 und dem maximalen Kontingent für den Durchsatz pro Sekunde für Ihr Konto liegen (standardmäßig 40.000).
- `scaling_policy`— Amazon Keyspaces unterstützt die Richtlinie zur Zielverfolgung. Das Auto Scaling-Ziel ist die bereitgestellte Lesekapazität der Tabelle.
- `target_tracking_scaling_policy_configuration`— Um die Zielverfolgungsrichtlinie zu definieren, müssen Sie den Zielwert definieren. Weitere Informationen zu Target Tracking und Cooldown-Perioden finden Sie unter [Target Tracking Scaling Policies](#) im Application Auto Scaling User Guide.
- `target_value`— Die Zielauslastungsrate der Tabelle. Die auto Skalierung von Amazon Keyspaces stellt sicher, dass das Verhältnis zwischen verbrauchter Lesekapazität und bereitgestellter Lesekapazität bei oder nahe diesem Wert bleibt. Sie definieren `target_value` als Prozentsatz. Ein Doppelter zwischen 20 und 90. (Erforderlich)
- `scale_in_cooldown`— Eine Abklingzeit in Sekunden zwischen den Skalierungsaktivitäten, sodass sich die Tabelle stabilisiert, bevor eine weitere Aktivitätsskala beginnt. Wenn kein Wert angegeben wird, ist der Standardwert 0. (Optional)
- `scale_out_cooldown`— Eine Abklingzeit in Sekunden zwischen Skalierungsaktivitäten, durch die sich die Tabelle stabilisiert, bevor eine weitere Scale-Out-Aktivität beginnt. Wenn kein Wert angegeben wird, ist der Standardwert 0. (Optional)
- `disable_scale_in`: Aboolean, das angibt, ob `scale-in` es für die Tabelle deaktiviert oder aktiviert ist. Dieser Parameter ist standardmäßig deaktiviert. Stellen Sie zum `scale-in` Einschalten den boolean Wert auf ein `FALSE`. Das bedeutet, dass die Lesekapazität für eine Tabelle in Ihrem Namen automatisch herunterskaliert wird. (Optional)
- `default_time_to_live`— Die Standardeinstellung für die Gültigkeitsdauer der Tabelle in Sekunden.
- `TAGS`— Eine Liste von Schlüssel-Wert-Paar-Tags, die bei ihrer Erstellung an die Ressource angehängt werden sollen.
- `clustering_order` besteht aus den folgenden Teilen:
  - `column_name`— Der Name der Spalte.

- *ASC* / *DESC*— Legt den Modifikator für die aufsteigende (*ASC*) oder absteigende (*DESC*) Reihenfolge fest. Wenn er nicht angegeben ist, ist die Standardreihenfolge *ASC*.

## Beispiel

```
CREATE TABLE IF NOT EXISTS "my_keyspace".my_table (
 id text,
 name text,
 region text,
 division text,
 project text,
 role text,
 pay_scale int,
 vacation_hrs float,
 manager_id text,
 PRIMARY KEY (id,division))
WITH CUSTOM_PROPERTIES={
 'capacity_mode':{
 'throughput_mode':
'PROVISIONED', 'read_capacity_units': 10, 'write_capacity_units': 20
 },
 'point_in_time_recovery':{'status':
'enabled'}},
 'encryption_specification':{
 'encryption_type':
'CUSTOMER_MANAGED_KMS_KEY',

 'kms_key_idenfierier':'arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
}
AND CLUSTERING ORDER BY (division ASC)
AND TAGS={'key1':'val1', 'key2':'val2'}
AND default_time_to_live = 3024000;
```

In einer Tabelle, die Clusterspalten verwendet, können Spalten, die keine Clusterbildung bilden, in der Tabellendefinition als statisch deklariert werden. Weitere Hinweise zu statischen Spalten finden Sie unter [the section called “Statische Spalten”](#)

## Beispiel

```
CREATE TABLE "my_keyspace".my_table (
```

```
id int,
name text,
region text,
division text,
project text STATIC,
PRIMARY KEY (id,division));
```

## ALTER TABLE

Verwenden Sie die ALTER TABLE Anweisung, um neue Spalten hinzuzufügen, Tags hinzuzufügen oder die benutzerdefinierten Eigenschaften der Tabelle zu ändern.

### Syntax

```
alter_table_statement ::= ALTER TABLE table_name

 [ADD (column_definition | column_definition_list)]
 [[ADD | DROP] TAGS {'key1':'val1', 'key2':'val2'}]
 [WITH table_options [, ...]] ;

column_definition ::= column_name cql_type
```

Wobei gilt:

- *table\_name* ist der Name der Tabelle, die geändert werden soll.
- *column\_definition* ist der Name der Spalte und des Datentyps, die hinzugefügt werden sollen.
- *column\_definition\_list* ist eine durch Kommas getrennte Liste von Spalten, die in Klammern stehen.
- *table\_options* besteht aus den folgenden Komponenten:
  - *CUSTOM\_PROPERTIES*— Eine Karte mit Einstellungen, die für Amazon Keyspaces spezifisch sind.
  - *capacity\_mode*: Gibt den Lese-/Schreibdurchsatzkapazitätsmodus für die Tabelle an. Die Optionen sind `throughput_mode:PAY_PER_REQUEST` und `throughput_mode:PROVISIONED`. Der bereitgestellte Kapazitätsmodus erfordert `read_capacity_units` und `write_capacity_units` als Eingaben. Der Standardwert ist `throughput_mode:PAY_PER_REQUEST`.

- `client_side_timestamps`: Gibt an, ob clientseitige Zeitstempel für die Tabelle aktiviert oder deaktiviert sind. Die Optionen sind `{'status': 'enabled'}` und `{'status': 'disabled'}`. Wenn es nicht angegeben ist, ist die Standardeinstellung `status:disabled`. Nachdem clientseitige Zeitstempel für eine Tabelle aktiviert wurden, kann diese Einstellung nicht deaktiviert werden.
- `encryption_specification`: Gibt die Verschlüsselungsoption für die Verschlüsselung im Ruhezustand an. Die Optionen sind `encryption_type:AWS_OWNED_KMS_KEY` und `encryption_type:CUSTOMER_MANAGED_KMS_KEY`. Die vom Kunden verwaltete Verschlüsselungsoption erfordert den AWS KMS Schlüssel im Format Amazon Resource Name (ARN) als Eingabe: `kms_key_identifizier:ARN`.
- `point_in_time_recovery`: Gibt an, ob die point-in-time Wiederherstellung für die Tabelle aktiviert oder deaktiviert ist. Die Optionen sind `status:enabled` und `status:disabled`. Der Standardwert ist `status:disabled`.
- `replica_updates`: Gibt die AWS-Region spezifischen Einstellungen einer Tabelle mit mehreren Regionen an. Bei einer Tabelle mit mehreren Regionen können Sie die Lesekapazität der Tabelle je nach Region unterschiedlich konfigurieren. AWS-Region Sie können dies tun, indem Sie die folgenden Parameter konfigurieren. Weitere Informationen und Beispiele finden Sie unter [the section called “Aktualisieren der bereitgestellten Kapazität und Auto-Scaling-Einstellungen einer Tabelle mit mehreren Regionen \(CQL\)”](#).
  - `region`— Das AWS-Region des Tabellenreplikats mit den folgenden Einstellungen:
    - `read_capacity_units`
- `t1`: Aktiviert die benutzerdefinierten Time to Live-Einstellungen für die Tabelle. Verwenden Sie zum Aktivieren `status:enabled`. Der Standardwert ist `status:disabled`. Nachdem `t1` es aktiviert wurde, können Sie es für die Tabelle nicht mehr deaktivieren.
- `AUTOSCALING_SETTINGS` beinhaltet die optionalen Auto Scaling-Einstellungen für bereitgestellte Tabellen. Die Syntax und ausführliche Beschreibungen finden Sie unter [the section called “CREATE TABLE”](#). Beispiele finden Sie unter [the section called “Aktivieren der automatischen Skalierung für eine vorhandene Tabelle mithilfe von CQL”](#).
- `default_time_to_live`: Die Standardeinstellung für die Gültigkeitsdauer der Tabelle in Sekunden.
- `TAGS` ist eine Liste von Schlüssel-Wert-Paar-Tags, die an die Ressource angehängt werden sollen.

**Note**

Mit ALTER TABLE können Sie nur eine einzige benutzerdefinierte Eigenschaft ändern. Sie können nicht mehr als einen ALTER TABLE-Befehl in derselben Anweisung kombinieren.

**Beispiele**

Die folgende Anweisung zeigt, wie einer vorhandenen Tabelle eine Spalte hinzugefügt wird.

```
ALTER TABLE mykeyspace.mytable ADD (ID int);
```

Diese Anweisung zeigt, wie zwei Sammlungsspalten zu einer vorhandenen Tabelle hinzugefügt werden:

- Eine eingefrorene Sammlungsspalte *col\_frozen\_list*, die eine verschachtelte eingefrorene Sammlung enthält
- Eine nicht eingefrorene Sammlungsspalte *col\_map*, die eine verschachtelte eingefrorene Sammlung enthält

```
ALTER TABLE my_Table ADD(col_frozen_list FROZEN<LIST<FROZEN<SET<TEXT>>>>, col_map MAP<INT, FROZEN<SET<INT>>>>);
```

Um den Kapazitätsmodus einer Tabelle zu ändern und Lese- und Schreibkapazitätseinheiten anzugeben, können Sie die folgende Anweisung verwenden.

```
ALTER TABLE mykeyspace.mytable WITH CUSTOM_PROPERTIES={'capacity_mode': {'throughput_mode': 'PROVISIONED', 'read_capacity_units': 10, 'write_capacity_units': 20}};
```

Die folgende Anweisung gibt einen vom Kunden verwalteten KMS-Schlüssel für die Tabelle an.

```
ALTER TABLE mykeyspace.mytable WITH CUSTOM_PROPERTIES={
 'encryption_specification':{
 'encryption_type': 'CUSTOMER_MANAGED_KMS_KEY',
 'kms_key_identifier': 'arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
}
```

```
};
```

Um die point-in-time Wiederherstellung für eine Tabelle zu aktivieren, können Sie die folgende Anweisung verwenden.

```
ALTER TABLE mykeyspace.mytable WITH CUSTOM_PROPERTIES={'point_in_time_recovery':
{'status': 'enabled'}};
```

Um einen Standardwert für die Gültigkeitsdauer in Sekunden für eine Tabelle festzulegen, können Sie die folgende Anweisung verwenden.

```
ALTER TABLE my_table WITH default_time_to_live = 2592000;
```

Diese Anweisung ermöglicht benutzerdefinierte Time-to-Live-Einstellungen für eine Tabelle.

```
ALTER TABLE mytable WITH CUSTOM_PROPERTIES={'ttl':{'status': 'enabled'}};
```

## TABELLE WIEDERHERSTELLEN

Verwenden Sie die `RESTORE TABLE` Anweisung, um eine Tabelle auf einen bestimmten Zeitpunkt zurückzusetzen. Für diese Anweisung muss die point-in-time Wiederherstellung für eine Tabelle aktiviert sein. Weitere Informationen finden Sie unter [oint-in-time P-Wiederherstellung](#).


### Syntax

```
restore_table_statement ::=
 RESTORE TABLE restored_table_name FROM TABLE source_table_name
 [WITH table_options [, ...]];
```

Wobei gilt:

- *restored\_table\_name* ist der Name der wiederhergestellten Tabelle.
- *source\_table\_name* ist der Name der Quelltable.
- *table\_options* besteht aus den folgenden Teilen:
  - *restore\_timestamp* ist die Uhrzeit des Wiederherstellungspunkts im Format ISO 8601. Wenn er nicht angegeben ist, wird der aktuelle Zeitstempel verwendet.
  - *CUSTOM\_PROPERTIES*— Eine Karte mit Einstellungen, die für Amazon Keyspaces spezifisch sind.

- `capacity_mode`: Gibt den Lese-/Schreibdurchsatzkapazitätsmodus für die Tabelle an. Die Optionen sind `throughput_mode:PAY_PER_REQUEST` und `throughput_mode:PROVISIONED`. Der bereitgestellte Kapazitätsmodus erfordert `read_capacity_units` und `write_capacity_units` als Eingaben. Die Standardeinstellung ist die aktuelle Einstellung aus der Quelltable.
- `encryption_specification`: Gibt die Verschlüsselungsoption für die Verschlüsselung im Ruhezustand an. Die Optionen sind `encryption_type:AWS_OWNED_KMS_KEY` und `encryption_type:CUSTOMER_MANAGED_KMS_KEY`. Die vom Kunden verwaltete Verschlüsselungsoption erfordert den AWS KMS Schlüssel im Format Amazon Resource Name (ARN) als Eingabe: `kms_key_identifier:ARN`. Um eine mit einem vom Kunden verwalteten Schlüssel verschlüsselte Tabelle in eine mit einem verschlüsselte Tabelle wiederherzustellen AWS-eigener Schlüssel, benötigt Amazon Keyspaces Zugriff auf den AWS KMS Schlüssel der Quelltable.
- `point_in_time_recovery`: Gibt an, ob point-in-time die Wiederherstellung für die Tabelle aktiviert oder deaktiviert ist. Die Optionen sind `status:enabled` und `status:disabled`. Anders als beim Erstellen neuer Tabellen ist der Standardstatus für wiederhergestellte Tabellen darauf `status:enabled` zurückzuführen, dass die Einstellung von der Quelltable übernommen wird. Um PITR für wiederhergestellte Tabellen zu deaktivieren, müssen Sie `status:disabled` dies explizit festlegen.
- `replica_updates`: Gibt die AWS-Region spezifischen Einstellungen einer Tabelle mit mehreren Regionen an. Bei einer Tabelle mit mehreren Regionen können Sie die Lesekapazität der Tabelle je nach Region unterschiedlich konfigurieren. AWS-Region Sie können dies tun, indem Sie die folgenden Parameter konfigurieren.
  - `region`— Das AWS-Region des Tabellenreplikats mit den folgenden Einstellungen:
    - `read_capacity_units`
- `AUTOSCALING_SETTINGS` beinhaltet die optionalen Auto Scaling-Einstellungen für bereitgestellte Tabellen. Eine ausführliche Syntax und Beschreibungen finden Sie unter [the section called "CREATE TABLE"](#).
- `TAGS` ist eine Liste von Schlüssel-Wert-Paar-Tags, die an die Ressource angehängt werden sollen.

 Note

Gelöschte Tabellen können nur zum Zeitpunkt des Löschvorgangs wiederhergestellt werden.

## Beispiel

```
RESTORE TABLE mykeyspace.mytable_restored from table mykeyspace.my_table
WITH restore_timestamp = '2020-06-30T04:05:00+0000'
AND custom_properties = {'point_in_time_recovery':{'status':'disabled'},
 'capacity_mode':{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 10,
 'write_capacity_units': 20}}
AND TAGS={'key1':'val1', 'key2':'val2'};
```

## DROP TABLE

Verwenden Sie die `DROP TABLE` Anweisung, um eine Tabelle aus dem Schlüsselraum zu entfernen.

### Syntax

```
drop_table_statement ::=
 DROP TABLE [IF EXISTS] table_name
```

Wobei gilt:

- `IF EXISTS` `DROP TABLE` verhindert, dass ein Fehler auftritt, wenn die Tabelle nicht existiert. (Optional)
- *table\_name* ist der Name der Tabelle, die gelöscht werden soll.

## Beispiel

```
DROP TABLE "myGSGKeyspace".employees_tbl;
```

# DML-Anweisungen (Datenmanipulationssprache) in Amazon Keyspaces

Data Manipulation Language (DML) ist der Satz von Cassandra Query Language (CQL) - Anweisungen, mit denen Sie Daten in Amazon Keyspaces-Tabellen (für Apache Cassandra) verwalten. Sie verwenden DML-Anweisungen, um Daten in einer Tabelle hinzuzufügen, zu ändern oder zu löschen.

Sie verwenden DML-Anweisungen auch, um Daten in einer Tabelle abzufragen. (Beachten Sie, dass CQL keine Verknüpfungen oder Unterabfragen unterstützt.)



## Themen

- [SELECT](#)
- [INSERT](#)
- [UPDATE](#)
- [DELETE](#)

## SELECT

Verwenden Sie eine SELECT-Anweisung, um Daten abzufragen.

### Syntax

```

select_statement ::= SELECT [JSON] (select_clause | '*')
 FROM table_name
 [WHERE 'where_clause']
 [ORDER BY 'ordering_clause']
 [LIMIT (integer | bind_marker)]
 [ALLOW FILTERING]

select_clause ::= selector [AS identifier] (',' selector [AS identifier])
selector ::= column_name
 | term
 | CAST '(' selector AS cql_type ')'
 | function_name '(' [selector (',' selector)*] ')'

where_clause ::= relation (AND relation)*
relation ::= column_name operator term
 TOKEN

operator ::= '=' | '<' | '>' | '<=' | '>=' | IN | CONTAINS | CONTAINS KEY
ordering_clause ::= column_name [ASC | DESC] (',' column_name [ASC | DESC])*

```

### Beispiele

```

SELECT name, id, manager_id FROM "myGSGKeyspace".employees_tbl ;

SELECT JSON name, id, manager_id FROM "myGSGKeyspace".employees_tbl ;

```

Eine Tabelle, die JSON-kodierte Datentypen Amazon Keyspaces-Datentypen zuordnet, finden Sie unter [the section called “JSON-Kodierung von Amazon Keyspaces-Datentypen”](#)

## Verwenden des Schlüsselworts **IN**

Das IN Schlüsselwort gibt die Gleichheit für einen oder mehrere Werte an. Es kann auf den Partitionsschlüssel und die Clusterspalte angewendet werden. Die Ergebnisse werden in der Reihenfolge zurückgegeben, in der die Schlüssel in der SELECT Anweisung dargestellt werden.

### Beispiele

```
SELECT * from mykeyspace.mytable WHERE primary.key1 IN (1,2) and clustering.key1 = 2;
SELECT * from mykeyspace.mytable WHERE primary.key1 IN (1,2) and clustering.key1 <= 2;
SELECT * from mykeyspace.mytable WHERE primary.key1 = 1 and clustering.key1 IN (1, 2);
SELECT * from mykeyspace.mytable WHERE primary.key1 <= 2 and clustering.key1 IN (1, 2)
ALLOW FILTERING;
```

Weitere Informationen zum IN Schlüsselwort und zur Verarbeitung der Anweisung durch Amazon Keyspaces finden Sie unter [the section called “INSELECTAussage”](#).

### Ergebnisse bestellen

Die ORDER BY Klausel gibt die Sortierreihenfolge der zurückgegebenen Ergebnisse an. Sie verwendet als Argumente eine Liste von Spaltennamen zusammen mit der Sortierreihenfolge für jede Spalte. Sie können Clusterspalten nur in Sortierklauseln angeben. Spalten, bei denen es sich nicht um Gruppierungen handelt, sind nicht zulässig. Die Optionen für die Sortierreihenfolge gelten ASC für die aufsteigende und DESC für die absteigende Sortierreihenfolge. Wenn die Sortierreihenfolge weggelassen wird, wird die Standardreihenfolge der Clusterspalte verwendet. Mögliche Sortierreihenfolgen finden Sie unter [the section called “Ergebnisse bestellen”](#).

### Beispiel

```
SELECT name, id, division, manager_id FROM "myGSGKeyspace".employees_tbl WHERE id =
'012-34-5678' ORDER BY division;
```

Bei Verwendung ORDER BY mit dem IN Schlüsselwort werden die Ergebnisse innerhalb einer Seite angeordnet. Eine vollständige Neuordnung mit deaktivierter Paginierung wird nicht unterstützt.

## TOKEN

Sie können die TOKEN Funktion auf die PARTITION KEY Spalten in SELECT und WHERE -Klauseln anwenden. Mit dieser TOKEN Funktion gibt Amazon Keyspaces Zeilen zurück, die auf dem

zugewiesenen Token-Wert von basieren `PARTITION_KEY` und nicht auf dem Wert von `PARTITION_KEY`

`TOKEN`Relationen werden mit dem Schlüsselwort nicht unterstützt. `IN`

## Beispiele

```
SELECT TOKEN(id) from my_table;

SELECT TOKEN(id) from my_table WHERE TOKEN(id) > 100 and TOKEN(id) < 10000;
```

## TTL-Funktion

Sie können die `TTL` Funktion mit der `SELECT` Anweisung verwenden, um die für eine Spalte gespeicherte Ablaufzeit in Sekunden abzurufen. Wenn kein `TTL` Wert festgelegt ist, kehrt die Funktion zurücknull.

## Beispiel

```
SELECT TTL(my_column) from my_table;
```

Die `TTL` Funktion kann nicht für Spalten mit mehreren Zellen wie Sammlungen verwendet werden.

## WRITETIMEFunktion

Sie können die `WRITETIME` Funktion mit der `SELECT` Anweisung nur dann verwenden, um den Zeitstempel abzurufen, der als Metadaten für den Wert einer Spalte gespeichert ist, wenn die Tabelle clientseitige Zeitstempel verwendet. Weitere Informationen finden Sie unter [ClientZeitstempel](#).

```
SELECT WRITETIME(my_column) from my_table;
```

Die `WRITETIME` Funktion kann nicht für Spalten mit mehreren Zellen wie Sammlungen verwendet werden.

### Note

Aus Gründen der Kompatibilität mit dem gängigen Verhalten von Cassandra-Treibern werden tagbasierte Autorisierungsrichtlinien nicht durchgesetzt, wenn Sie mithilfe von Cassandra Query Language (CQL) -API-Aufrufen über Cassandra-Treiber und Entwicklertools

Operationen an Systemtabellen ausführen. Weitere Informationen finden Sie unter [the section called “ Amazon Keyspaces-Ressourcenzugriff basierend auf Tags”](#).

## INSERT

Verwenden Sie die INSERT Anweisung, um einer Tabelle eine Zeile hinzuzufügen.

### Syntax

```
insert_statement ::= INSERT INTO table_name (names_values | json_clause)
 [IF NOT EXISTS]
 [USING update_parameter (AND update_parameter)*]
names_values ::= names VALUES tuple_literal
json_clause ::= JSON string [DEFAULT (NULL | UNSET)]
names ::= '(' column_name (',' column_name)* ')'
```

### Beispiel

```
INSERT INTO "myGSGKeyspace".employees_tbl (id, name, project, region, division, role,
pay_scale, vacation_hrs, manager_id)
VALUES ('012-34-5678', 'Russ', 'NightFlight', 'US', 'Engineering', 'IC', 3, 12.5,
'234-56-7890') ;
```

### Parameter aktualisieren

INSERT unterstützt die folgenden Werte als `update_parameter`:

- **TTL**— Ein Zeitwert in Sekunden. Der maximal konfigurierbare Wert ist 630 720 000 Sekunden, was 20 Jahren entspricht.
- **TIMESTAMP**— Ein `bigint` Wert, der die Anzahl der Mikrosekunden seit der Standardbasiszeit darstellt, die `alsepoch`: 1. Januar 1970 um 00:00:00 Uhr GMT bekannt ist. Ein Zeitstempel in Amazon Keyspaces muss zwischen 2 Tagen in der Vergangenheit und 5 Minuten in der future liegen.

### Beispiel

```
INSERT INTO my_table (userid, time, subject, body, user)
```

```
VALUES (B79CB3BA-745E-5D9A-8903-4A02327A7E09, 96a29100-5e25-11ec-90d7-
b5d91eceda0a, 'Message', 'Hello', '205.212.123.123')
USING TTL 259200;
```

## JSON-Unterstützung

Eine Tabelle, die JSON-kodierte Datentypen Amazon Keyspaces-Datentypen zuordnet, finden Sie unter [the section called “JSON-Kodierung von Amazon Keyspaces-Datentypen”](#)

Sie können das JSON Schlüsselwort verwenden, um eine JSON -kodierte Map als einzelne Zeile einzufügen. Verwenden Sie für Spalten, die in der Tabelle vorhanden sind, aber in der JSON-Einfügeanweisung weggelassen werden, DEFAULT UNSET um die vorhandenen Werte beizubehalten. Wird verwendet DEFAULT NULL, um in jede Zeile ausgelassener Spalten einen NULL-Wert zu schreiben und die vorhandenen Werte zu überschreiben (es fallen die üblichen Schreibgebühren an). DEFAULT NULL ist die Standardoption.

## Beispiel

```
INSERT INTO "myGSGKeyspace".employees_tbl JSON '{"id":"012-34-5678",
 "name": "Russ",
 "project": "NightFlight",
 "region": "US",
 "division": "Engineering",
 "role": "IC",
 "pay_scale": 3,
 "vacation_hrs": 12.5,
 "manager_id": "234-56-7890"}';
```

Wenn die JSON-Daten doppelte Schlüssel enthalten, speichert Amazon Keyspaces den letzten Wert für den Schlüssel (ähnlich wie Apache Cassandra). Im folgenden Beispiel, in dem sich der doppelte Schlüssel befindet `id`, wird der Wert verwendet `234-56-7890`.

## Beispiel

```
INSERT INTO "myGSGKeyspace".employees_tbl JSON '{"id":"012-34-5678",
 "name": "Russ",
 "project": "NightFlight",
 "region": "US",
 "division": "Engineering",
 "role": "IC",
 "pay_scale": 3,
```

```
"vacation_hrs": 12.5,
"id": "234-56-7890"}';
```

## UPDATE

Verwenden Sie die UPDATE Anweisung, um eine Zeile in einer Tabelle zu ändern.

### Syntax

```
update_statement ::= UPDATE table_name
 [USING update_parameter (AND update_parameter)*]
 SET assignment (',' assignment)*
 WHERE where_clause
 [IF (EXISTS | condition (AND condition)*)]
update_parameter ::= (integer | bind_marker)
assignment ::= simple_selection '=' term
 | column_name '=' column_name ('+' | '-') term
 | column_name '=' list_literal '+' column_name
simple_selection ::= column_name
 | column_name '[' term ']'
 | column_name '.' `field_name`
condition ::= simple_selection operator term
```

### Beispiel

```
UPDATE "myGSGKeyspace".employees_tbl SET pay_scale = 5 WHERE id = '567-89-0123' AND
division = 'Marketing' ;
```

Verwenden Sie die folgende Syntaxcounter, um a zu inkrementieren. Weitere Informationen finden Sie unter [the section called “Zähler”](#).

```
UPDATE ActiveUsers SET counter = counter + 1 WHERE user = A70FE1C0-5408-4AE3-
BE34-8733E5K09F14 AND action = 'click';
```

### Parameter aktualisieren

UPDATE unterstützt die folgenden Werte als `update_parameter`:

- TTL— Ein Zeitwert in Sekunden. Der maximal konfigurierbare Wert ist 630 720 000 Sekunden, was 20 Jahren entspricht.

- **TIMESTAMP**— Ein `bigint` Wert, der die Anzahl der Mikrosekunden seit der Standardbasiszeit darstellt, die `alsepoch`: 1. Januar 1970 um 00:00:00 Uhr GMT bekannt ist. Ein Zeitstempel in Amazon Keyspaces muss zwischen 2 Tagen in der Vergangenheit und 5 Minuten in der future liegen.

### Beispiel

```
UPDATE my_table (userid, time, subject, body, user)
 VALUES (B79CB3BA-745E-5D9A-8903-4A02327A7E09, 96a29100-5e25-11ec-90d7-
b5d91eceda0a, 'Message', 'Hello again', '205.212.123.123')
 USING TIMESTAMP '2022-11-03 13:30:54+0400';
```

## DELETE

Verwenden Sie die `DELETE` Anweisung, um eine Zeile aus einer Tabelle zu entfernen.

### Syntax

```
delete_statement ::= DELETE [simple_selection (',' simple_selection)]
 FROM table_name
 [USING update_parameter (AND update_parameter)*]
 WHERE where_clause
 [IF (EXISTS | condition (AND condition)*)]

simple_selection ::= column_name
 | column_name '[' term ']'
 | column_name '.' `field_name`

condition ::= simple_selection operator term
```

Wobei gilt:

- `table_name` ist die Tabelle, die die Zeile enthält, die Sie löschen möchten.

### Beispiel

```
DELETE manager_id FROM "myGSGKeyspace".employees_tbl WHERE id='789-01-2345' AND
 division='Executive' ;
```

`DELETE` unterstützt den folgenden Wert als `update_parameter`:

- **TIMESTAMP**— Ein `bigint` Wert, der die Anzahl der Mikrosekunden seit der Standardbasiszeit darstellt, die `alsepoch`: 1. Januar 1970 um 00:00:00 GMT bekannt ist.

## Integrierte Funktionen in Amazon Keyspaces

Amazon Keyspaces (für Apache Cassandra) unterstützt eine Vielzahl integrierter Funktionen, die Sie in Cassandra Query Language (CQL) -Anweisungen verwenden können.

Themen

- [Skalarfunktionen](#)

### Skalarfunktionen

Eine Skalarfunktion führt eine Berechnung für einen einzelnen Wert durch und gibt das Ergebnis als einzelnen Wert zurück. Amazon Keyspaces unterstützt die folgenden Skalarfunktionen.

| Funktion                      | Beschreibung                                                                      |
|-------------------------------|-----------------------------------------------------------------------------------|
| <code>blobAsType</code>       | Gibt einen Wert des angegebenen Datentyps zurück.                                 |
| <code>cast</code>             | Konvertiert einen systemeigenen Datentyp in einen anderen systemeigenen Datentyp. |
| <code>currentDate</code>      | Gibt das aktuelle Datum/die aktuelle Uhrzeit als Datum zurück.                    |
| <code>currentTime</code>      | Gibt das aktuelle Datum/die aktuelle Uhrzeit als Uhrzeit zurück.                  |
| <code>currentTimestamp</code> | Gibt das aktuelle Datum/die aktuelle Uhrzeit als Zeitstempel zurück.              |
| <code>currentTimeUUID</code>  | Gibt das aktuelle Datum/die aktuelle Uhrzeit als zurück. <code>timeuuid</code>    |
| <code>fromJson</code>         | Konvertiert die JSON-Zeichenfolge in den Datentyp der ausgewählten Spalte.        |



| Funktion                     | Beschreibung                                                                                                                                                              |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>maxTimeuuid</code>     | Gibt den größtmöglichen Wert <code>timeuuid</code> für Zeitstempel oder Datumszeichenfolge zurück.                                                                        |
| <code>minTimeuuid</code>     | Gibt den kleinstmöglichen Wert <code>timeuuid</code> für einen Zeitstempel oder eine Datumszeichenfolge zurück.                                                           |
| <code>now</code>             | Gibt ein neues Unikat <code>timeuuid</code> zurück. Wird für INSERT UPDATE , - und DELETE Anweisungen sowie als Teil der WHERE Klausel in SELECT Anweisungen unterstützt. |
| <code>toDate</code>          | Konvertiert entweder einen Zeitstempel <code>timeuuid</code> oder einen Zeitstempel in einen Datumstyp.                                                                   |
| <code>toJson</code>          | Gibt den Spaltenwert der ausgewählten Spalte im JSON-Format zurück.                                                                                                       |
| <code>token</code>           | Gibt den Hashwert des Partitionsschlüssels zurück.                                                                                                                        |
| <code>toTimestamp</code>     | Konvertiert entweder ein <code>timeuuid</code> oder ein Datum in einen Zeitstempel.                                                                                       |
| <code>TTL</code>             | Gibt die Ablaufzeit in Sekunden für eine Spalte zurück.                                                                                                                   |
| <code>typeAsBlob</code>      | Konvertiert den angegebenen Datentyp in <code>blob</code> .                                                                                                               |
| <code>toUnixTimestamp</code> | Konvertiert entweder einen Zeitstempel <code>timeuuid</code> oder einen Zeitstempel in <code>bigint</code> .                                                              |

| Funktion        | Beschreibung                                                                                                                                                     |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| uuid            | Gibt eine zufällige Version 4-UUID zurück. Wird für INSERT UPDATE , - und DELETE Anweisungen sowie als Teil der WHERE Klausel in SELECT Anweisungen unterstützt. |
| writetime       | Gibt den Zeitstempel des Werts der angegebenen Spalte zurück.                                                                                                    |
| dateOf          | (Veraltet) Extrahiert den Zeitstempel von a timeuuid und gibt den Wert als Datum zurück.                                                                         |
| unixTimestampOf | (Veraltet) Extrahiert den Zeitstempel von a und gibt den Wert als timeuuid rohen 64-Bit-Ganzzahl-Zeitstempel zurück.                                             |

# Kontingente für Amazon Keyspaces (für Apache Cassandra)

In diesem Abschnitt werden aktuelle Kontingente und Standardwerte für Amazon Keyspaces (für Apache Cassandra) beschrieben.

## Themen

- [Service Quotas für Amazon Keyspaces](#)
- [Erhöhen oder Verringern des Durchsatzes \(für Tabellen im Modus bereitgestellter Kapazität\)](#)
- [Amazon Keyspaces-Verschlüsselung im Ruhezustand](#)

## Service Quotas für Amazon Keyspaces

Die folgende Tabelle enthält Amazon Keyspaces (für Apache Cassandra)-Kontingente und die Standardwerte. Informationen darüber, welche Kontingente angepasst werden können, finden Sie in der [Service Quotas](#)-Konsole, in der Sie auch Kontingenterhöhungen anfordern können. Weitere Informationen zu Kontingenten erhalten Sie unter AWS Support.

| Kontingent                        | Beschreibung                                                                                                                                                                  | Standardeinstellung für Amazon Keyspaces |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Maximale Keyspaces pro AWS-Region | Die maximale Anzahl von Keyspaces für diesen Abonnenten pro Region. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.                   | 256                                      |
| Maximale Tabellen pro AWS-Region  | Die maximale Anzahl von Tabellen für alle Keyspaces für diesen Abonnenten pro Region. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen. | 256                                      |

| Kontingent                                                     | Beschreibung                                                                                                                                                                                                                                                                                                             | Standardeinstellung für Amazon Keyspaces |
|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Maximale Tabellenschemagröße                                   | Die maximale Größe eines Tabellenschemas.                                                                                                                                                                                                                                                                                | 350 KB                                   |
| Max. gleichzeitige DDL-Operationen                             | Die maximale Anzahl gleichzeitiger DDL-Operationen, die für diesen Abonnenten pro Region zulässig sind.                                                                                                                                                                                                                  | 50                                       |
| Maximale Abfragen pro Verbindung                               | Die maximale Anzahl von CQL-Abfragen, die pro Sekunde von einer einzelnen Client-TCP-Verbindung verarbeitet werden können.                                                                                                                                                                                               | 3000                                     |
| Maximale Zeilengröße                                           | Die maximale Größe einer Zeile, ausgenommen statische Spaltendaten. Details hierzu finden Sie unter <a href="#">the section called “Berechnung der Zeilengröße”</a> .                                                                                                                                                    | 1 MB                                     |
| Maximale Anzahl von Spalten in - INSERT und -UPDATEAnweisungen | Die maximale Anzahl von Spalten, die in CQL- INSERT oder -UPDATEAnweisungen zulässig sind. Eine - INSERT oder -UPDATEAnweisung unterstützt bis zu 225 reguläre Spalten, wenn Time to Live (TTL) deaktiviert ist. Wenn TTL aktiviert ist, können bis zu 166 reguläre Spalten in einer einzigen Operation geändert werden. | 225/166                                  |

| Kontingent                                                              | Beschreibung                                                                                                                                                                                                                                                                                                                                                           | Standardeinstellung für Amazon Keyspaces |
|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Maximale statische Daten pro logischer Partition                        | Die maximale Aggregatgröße statischer Daten in einer logischen Partition. Details hierzu finden Sie unter <a href="#">the section called “Berechnen der statischen Spaltengröße pro logischer Partition”</a> .                                                                                                                                                         | 1 MB                                     |
| Maximale Unterabfragen pro IN SELECT Anweisung                          | Die maximale Anzahl von Unterabfragen, die Sie für das IN Schlüsselwort in einer SELECT Anweisung verwenden können. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.                                                                                                                                                            | 100                                      |
| Maximale Anzahl verschachtelter eingefrorener Sammlungen pro AWS-Region | Die maximale Anzahl von verschachtelten Sammlungen, die unterstützt werden, wenn Sie das FROZEN Schlüsselwort für eine Spalte mit einem Sammlungsdatentyp verwenden. Weitere Informationen zu eingefrorenen Sammlungen finden Sie unter <a href="#">the section called “Sammlungstypen”</a> . Um die Verschachtelungsebene zu erhöhen, wenden Sie sich an AWS Support. | 5                                        |

| Kontingent                                    | Beschreibung                                                                                                                                                                                                                                                                        | Standardeinstellung für Amazon Keyspaces |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Maximaler Lesedurchsatz pro Sekunde           | Der maximale Lesedurchsatz pro Sekunde – Leseanforderungseinheiten (RRUs) oder Lesekapazitätseinheiten (RCUs) –, der einer Tabelle pro Region zugewiesen werden kann. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.                       | 40 000                                   |
| Maximaler Schreibdurchsatz pro Sekunde        | Der maximale Schreibdurchsatz pro Sekunde – Schreibanforderungseinheiten (WRUs) oder Schreibkapazitätseinheiten (WCUs), die einer Tabelle pro Region zugewiesen werden können. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.              | 40 000                                   |
| Lesedurchsatz auf Kontoebene (bereitgestellt) | Die maximale Anzahl von aggregierten Lesekapazitätseinheiten (RCUs), die dem Konto pro Region zugewiesen sind. Dies gilt nur für Tabellen im Modus bereitgestellter Lese-/Schreibkapazität. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen. | 80 000                                   |

| Kontingent                                              | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Standardeinstellung für Amazon Keyspaces |
|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Schreibdurchsatz auf Kontoebene (bereitgestellt)        | Die maximale Anzahl von aggregierten Schreibkapazitätseinheiten (WCU), die dem Konto pro Region zugewiesen sind. Dies gilt nur für Tabellen im Modus bereitgestellter Lese-/Schreibkapazität. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.                                                                                                                                                                                  | 80 000                                   |
| Maximale Anzahl skalierbarer Ziele pro Region und Konto | Die maximale Anzahl von skalierbaren Zielen für das Konto pro Region. Eine Amazon Keyspaces-Tabelle zählt als ein skalierbares Ziel, wenn Auto Scaling für Lesekapazität aktiviert ist, und als ein weiteres skalierbares Ziel, wenn Auto Scaling für Schreibkapazität aktiviert ist. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole für Application Auto Scaling anpassen, indem Sie Skalierbare Ziele für Amazon Keyspaces auswählen. | 1.500                                    |

| Kontingent                                                                                           | Beschreibung                                                                                                                                                                                                      | Standardeinstellung für Amazon Keyspaces |
|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Maximale Partitionsschlüssel-Größe                                                                   | Die maximale Größe des zusammengesetzten Partitionsschlüssels. Bis zu 3 Byte zusätzlicher Speicher werden für Metadaten zur Rohgröße jeder Cluster-Spalte hinzugefügt, die im Partitionsschlüssel enthalten sind. | 2048 Bytes                               |
| Maximale Clustering-Schlüssel-Größe                                                                  | Die maximale kombinierte Größe aller Cluster-Spalten. Bis zu 4 Byte zusätzlicher Speicher werden zur Rohgröße jeder Cluster-Spalte für Metadaten hinzugefügt.                                                     | 850 Byte                                 |
| Maximale Anzahl gleichzeitiger Tabellenwiederherstellungen mit Point-in-time Recovery (PITR)         | Die maximale Anzahl gleichzeitiger Tabellenwiederherstellungen mit PITR pro Subscriber beträgt 4. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.                         | 4                                        |
| Maximale Datenmenge, die mithilfe der point-in-time Wiederherstellung wiederhergestellt wurde (PITR) | Die maximale Größe der Daten, die mit PITR innerhalb von 24 Stunden wiederhergestellt werden können. Sie können diesen Standardwert in der <a href="#">Service Quotas</a> -Konsole anpassen.                      | 5 TB                                     |



# Erhöhen oder Verringern des Durchsatzes (für Tabellen im Modus bereitgestellter Kapazität)

## Erhöhen des bereitgestellten Durchsatzes

Sie können `ReadCapacityUnits` oder `WriteCapacityUnits` so oft wie nötig erhöhen, indem Sie die `-Konsole` oder die `-ALTER TABLE`Anweisung verwenden. Die neuen Einstellungen werden erst wirksam, wenn die `ALTER TABLE`-Operation abgeschlossen ist.

Sie können Ihre Kontingente pro Konto nicht überschreiten, wenn Sie bereitgestellte Kapazität hinzufügen. Und Sie können die bereitgestellte Kapazität für Ihre Tabellen beliebig erhöhen. Weitere Informationen zu diesen Kontingenten auf Kontoebene finden Sie im vorhergehenden Abschnitt, [the section called "Service Quotas für Amazon Keyspaces"](#).

## Senken des bereitgestellten Durchsatzes

Für jede Tabelle in einer `-ALTER TABLE`Anweisung können Sie `ReadCapacityUnits` oder `WriteCapacityUnits` (oder beides) verringern. Die neuen Einstellungen werden erst wirksam, wenn die `ALTER TABLE`-Operation abgeschlossen ist.

Eine Reduzierung ist jederzeit und bis zu viermal täglich möglich. Tage sind entsprechend der Universal Coordinated Time (UTC) definiert. Außerdem ist eine zusätzliche Verringerung zulässig, wenn in der vorangehenden Stunde keine Verringerung erfolgt ist. Dadurch wird effektiv die maximale Anzahl von Rückgängen an einem Tag auf 27 erhöht (4 Rückgänge in der ersten Stunde und 1 Rückgänge für jedes der nachfolgenden 1-Stunden-Fenster an einem Tag).

## Amazon Keyspaces-Verschlüsselung im Ruhezustand

Sie können die Verschlüsselungsoptionen zwischen einem AWS -eigenen AWS KMS Schlüssel und einem kundenverwalteten AWS KMS Schlüssel bis zu viermal innerhalb eines 24-Stunden-Fensters pro Tabelle ändern, beginnend mit der Erstellung der Tabelle. Wenn in den vorhergehenden 6 Stunden kein Wechsel erfolgt ist, ist ein zusätzlicher Wechsel erlaubt. Dadurch wird die maximale Anzahl von Änderungen an einem Tag auf acht erhöht (vier Änderungen in den ersten sechs Stunden und eine Änderung für jedes der nachfolgenden sechsstündigen Fenster an einem Tag).

Sie können die Verschlüsselungsoption ändern, um einen - AWS eigenen AWS KMS Schlüssel so oft wie nötig zu verwenden, auch wenn das frühere Kontingent erschöpft ist.

Es gelten die folgenden Kontingente, sofern Sie keine höheren Werte beantragen. Informationen zum Anfordern einer Erhöhung des Servicekontingents finden Sie unter [AWS Support](#).

# Dokumentenverlauf für Amazon Keyspaces (für Apache Cassandra)

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der letzten Version von Amazon Keyspaces (für Apache Cassandra) beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

- Letzte Aktualisierung der Dokumentation: 7. Februar 2024

| Änderung                                                                                                | Beschreibung                                                                                                                       | Datum             |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Stellen Sie über Amazon Elastic Kubernetes Service eine Connect zu Amazon Keyspaces her</a> | Sie können jetzt einem step-by-step Tutorial folgen, um von Amazon EKS aus eine Verbindung zu Amazon Keyspaces herzustellen.       | 7. Februar 2024   |
| <a href="#">auto Skalierungs-APIs von Amazon Keyspaces für bereitgestellte Tabellen</a>                 | Amazon Keyspaces bietet CQL jetzt AWS API-Unterstützung für die Einrichtung von Auto Scaling mit bereitgestelltem Kapazitätsmodus. | 23. Januar 2024   |
| <a href="#">Unterstützung von Amazon Keyspaces Multiregion Replication für bereitgestellte Tabellen</a> | Amazon Keyspaces unterstützt jetzt den Modus „Bereitgestellte Kapazität“ für Tabellen mit mehreren Regionen.                       | 23. Januar 2024   |
| <a href="#">Amazon Keyspaces DML-Aktivität in Protokollen enthalten CloudTrail</a>                      | Sie können jetzt API-Aufrufe der Amazon Keyspaces Data Manipulation Language (DML) prüfen. AWS CloudTrail                          | 20. Dezember 2023 |
| <a href="#">Amazon Keyspaces-Unterstützung für das FROZEN Keyword</a>                                   | Amazon Keyspaces unterstützt jetzt das FROZEN Schlüssel                                                                            | 15. November 2023 |

wort für Sammlungsdatentypen.

[Aktualisierung der verwalteten Richtlinien von Amazon Keyspaces](#)

Amazon Keyspaces hat der `AmazonKeyspacesFullAccess` verwalteten Richtlinie neue Berechtigungen hinzugefügt, um Clients, die über VPC-Schnittstellen-Endpunkte eine Verbindung zu Amazon Keyspaces herstellen, Zugriff auf die Amazon EC2 EC2-Instance zu ermöglichen, um die `AmazonSystemPeers` Keyspaces-Tabelle mit Netzwerkinformationen aus der VPC zu aktualisieren.

3. Oktober 2023

[Aktualisierung der verwalteten Richtlinien von Amazon Keyspaces](#)

Amazon Keyspaces hat eine neue `AmazonKeyspacesReadOnlyAccess_v2` verwaltete Richtlinie erstellt, um Clients, die über VPC-Schnittstellen-Endpunkte eine Verbindung zu Amazon Keyspaces herstellen, den Zugriff auf die Amazon EC2 EC2-Instance zu ermöglichen, um die `AmazonSystemPeers` Keyspaces-Tabelle mit Netzwerkinformationen aus der VPC zu aktualisieren.

12. September 2023

|                                                                                          |                                                                                                                                                                                                                                                        |                |
|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <a href="#">Bewährte Methoden für das Erstellen von Verbindungen in Amazon Keyspaces</a> | Erfahren Sie, wie Sie die Client-Treiberkonfigurationen in Amazon Keyspaces verbessern und optimieren können.                                                                                                                                          | 30. Juni 2023  |
| <a href="#">Systemschlüsselräume sind jetzt für Amazon Keyspaces dokumentiert</a>        | Erfahren Sie, was in Systemschlüsselräumen gespeichert ist und wie Sie sie nach nützlichen Informationen in Amazon Keyspaces abfragen können.                                                                                                          | 21. Juni 2023  |
| <a href="#">Amazon Keyspaces unterstützt jetzt Multi-Region-Replikation</a>              | Amazon Keyspaces Multi-Region Replication unterstützt Sie bei der Wartung global verteilter Anwendungen, indem es Ihnen eine verbesserte Fehlertoleranz, Stabilität und Belastbarkeit bietet.                                                          | 5. Juni 2023   |
| <a href="#">Aktualisierung der verwalteten Richtlinien von Amazon Keyspaces</a>          | Amazon Keyspaces hat der AmazonKeyspacesFullAccess verwalteten Richtlinie neue Berechtigungen hinzugefügt, damit Amazon Keyspaces eine servicebezogene Rolle erstellen kann, wenn ein Administrator einen Schlüsselraum für mehrere Regionen erstellt. | 5. Juni 2023   |
| <a href="#">Amazon Keyspaces-Unterstützung für das IN Keyword</a>                        | Amazon Keyspaces unterstützt jetzt das IN Schlüsselwort in SELECT Kontoauszügen.                                                                                                                                                                       | 25. April 2023 |

|                                                                                                   |                                                                                                                                                                                                                                                                 |                   |
|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Kontoübergreifender Zugriff für Amazon Keyspaces und VPC-Schnittstellen-Endpunkte</a> | Erfahren Sie, wie Sie den kontoübergreifenden Zugriff für Amazon Keyspaces mit VPC-Endpunkten implementieren.                                                                                                                                                   | 20. April 2023    |
| <a href="#">Amazon Keyspaces-Unterstützung für clientseitige Zeitstempel</a>                      | Die clientseitigen Zeitstempel von Amazon Keyspace sind Cassandra-kompatible Zeitstempel auf Zellebene, die verteilten Anwendungen helfen, die Reihenfolge der Schreibvorgänge zu bestimmen, wenn verschiedene Clients Änderungen an denselben Daten vornehmen. | 14. März 2023     |
| <a href="#">Erste Schritte mit Amazon Keyspaces und Schnittstellen-VPC-Endpunkten</a>             | In diesem step-by-step Tutorial erfahren Sie, wie Sie von einer VPC aus eine Verbindung zu Amazon Keyspaces herstellen.                                                                                                                                         | 1. März 2023      |
| <a href="#">Optimierung der Kosten von Amazon Keyspaces-Tabellen</a>                              | Es stehen bewährte Methoden und Anleitungen zur Verfügung, mit denen Sie Strategien zur Kostenoptimierung Ihrer vorhandenen Amazon Keyspaces-Tabellen ermitteln können.                                                                                         | 17. Februar 2023  |
| <a href="#">Das Murmur3Partitioner ist jetzt die Standardinstellung</a>                           | Das Murmur3Partitioner ist jetzt die Standardpartitionierung in Amazon Keyspaces.                                                                                                                                                                               | 17. November 2022 |
| <a href="#">Amazon Keyspaces unterstützt jetzt Murmur3Partitioner</a>                             | Das Murmur3Partitioner ist jetzt in Amazon Keyspaces verfügbar.                                                                                                                                                                                                 | 9. November 2022  |

|                                                                                                       |                                                                                                                                                                                                                                    |                  |
|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Unterstützungsupdate für leere Zeichenketten und Blob-Werte</a>                           | Amazon Keyspaces unterstützt jetzt auch leere Zeichenketten und Blob-Werte als Cluster-Spaltenwerte.                                                                                                                               | 19. Oktober 2022 |
| <a href="#">Amazon Keyspaces ist jetzt verfügbar in AWS GovCloud (US)</a>                             | Amazon Keyspaces ist jetzt in der verfügbar AWS GovCloud (US) Region und unterliegt der FedRAMP-High-Konformität. <a href="#">Informationen zu verfügbaren Endpunkten finden Sie unter FIPS-Endgeräte.AWS GovCloud (US) Region</a> | 04. August 2022  |
| <a href="#">Überwachen Sie die Speicherkosten für Amazon Keyspaces-Tabellen mit Amazon CloudWatch</a> | Amazon Keyspaces hilft Ihnen jetzt dabei, die Speicherkosten für Tabellen mithilfe der <code>BillableTableSizeInBytes</code> CloudWatch Metrik im Laufe der Zeit zu überwachen und nachzuverfolgen.                                | 14. Juni 2022    |
| <a href="#">Amazon Keyspaces unterstützt jetzt Terraform</a>                                          | Sie können jetzt Terraform verwenden, um DDL-Operationen (Data Definition Language) in Amazon Keyspaces durchzuführen.                                                                                                             | 9. Juni 2022     |
| <a href="#">Unterstützung für Amazon token Keyspaces-Funktionen</a>                                   | Amazon Keyspaces hilft Ihnen jetzt, Anwendungsabfragen mithilfe der token Funktion zu optimieren.                                                                                                                                  | 19. April 2022   |

[Amazon Keyspaces-Integration mit Apache Spark](#)

Amazon Keyspaces hilft Ihnen jetzt, Daten in Apache Spark einfacher zu lesen und zu schreiben, indem Sie den [Open-Source-Spark Cassandra Connector](#) verwenden.

19. April 2022

[Amazon Keyspaces API-Referenz](#)

Amazon Keyspaces unterstützt Operationen auf Steuerungsebene zur Verwaltung von Schlüsselräumen und Tabellen mithilfe des AWS SDK und AWS CLI. Das API-Referenzhandbuch beschreibt die unterstützten Operationen auf der Kontrollebene ausführlich.

2. März 2022

[So beheben Sie häufig auftretende Konfigurationsprobleme bei der Verwendung von Amazon Keyspaces.](#)

Erfahren Sie mehr darüber, wie Sie häufig auftretende Konfigurationsprobleme lösen können, die bei der Verwendung von Amazon Keyspaces auftreten können.

22. November 2021

[Amazon Keyspaces-Unterstützung für Time to Live \(TTL\).](#)

Amazon Keyspaces Time to Live (TTL) hilft Ihnen dabei, Ihre Anwendungslogik zu vereinfachen und den Speicherpreis zu optimieren, indem Daten aus Tabellen automatisch ablaufen.

18. Oktober 2021



[Migrieren Sie Daten mit DSBulk zu Amazon Keyspaces.](#)

Ein tep-by-step Tutorial für die Migration von Daten von Apache Cassandra zu Amazon Keyspaces mit dem DataStax Bulk Loader (DSBulk).

9. August 2021

[Amazon Keyspaces-Unterstützung für VPC-Endpunkteinträge in der system.peers Tabelle.](#)

Amazon Keyspaces ermöglicht es Ihnen, die `system.peers` Tabelle mit verfügbaren VPC-Endpunktinformationen der Schnittstelle zu füllen, um den Lastenausgleich zu verbessern und den Lese-/Schreibdurchsatz zu erhöhen.

29. Juli 2021

[Aktualisierung der von IAM verwalteten Richtlinien zur Unterstützung von kundenverwalteten Schlüsseln. AWS KMS](#)

IAM-verwaltete Richtlinien für Amazon Keyspaces beinhalten jetzt Berechtigungen zum Auflisten und Anzeigen verfügbarer, von Kunden verwalteter AWS KMS Schlüssel, die in gespeichert sind. AWS KMS

1. Juni 2021

[Amazon Keyspaces-Unterstützung für vom Kunden verwaltete AWS KMS Schlüssel.](#)

Mit Amazon Keyspaces können Sie die Kontrolle über vom Kunden verwaltete AWS KMS Schlüssel übernehmen, die AWS KMS zur Verschlüsselung im Ruhezustand gespeichert sind.

1. Juni 2021

[Amazon Keyspaces-Unterstützung für JSON-Syntax](#)

Amazon Keyspaces hilft Ihnen dabei, JSON-Dokumente einfacher zu lesen und zu schreiben, indem es die JSON-Syntax für INSERT- und SELECT-Operationen unterstützt.

21. Januar 2021

[Amazon Keyspaces-Unterstützung für statische Spalten](#)

Amazon Keyspaces hilft Ihnen jetzt dabei, gemeinsame Daten zwischen mehreren Zeilen effizient zu aktualisieren und zu speichern, indem statische Spalten verwendet werden.

9. November 2020

[GA-Version der NoSQL Workbench-Unterstützung für Amazon Keyspaces](#)

NoSQL Workbench ist eine clientseitige Anwendung, mit der Sie nichtrelationale Datenmodelle für Amazon Keyspaces einfacher entwerfen und visualisieren können. NoSQL Workbench-Clients sind für Windows, macOS und Linux verfügbar.

28. Oktober 2020

[Vorabversion der NoSQL Workbench-Unterstützung für Amazon Keyspaces](#)

NoSQL Workbench ist eine clientseitige Anwendung, mit der Sie nichtrelationale Datenmodelle für Amazon Keyspaces einfacher entwerfen und visualisieren können. NoSQL Workbench-Clients sind für Windows, macOS und Linux verfügbar.

5. Oktober 2020

[Neue Codebeispiele für den programmatischen Zugriff auf Amazon Keyspaces](#)

Wir fügen weiterhin Codebeispiele für den programmatischen Zugriff auf Amazon Keyspaces hinzu. Beispiele sind jetzt für Java-, Python-, Go-, C#- und Perl-Cassandra-Treiber verfügbar, die Apache Cassandra Version 3.11.2 unterstützen.

17. Juli 2020

[point-in-time Wiederherstellung von Amazon Keyspaces \(PITR\)](#)

Amazon Keyspaces bietet jetzt point-in-time Recovery (PITR), um Ihre Tabellen vor versehentlichen Schreib- oder Löschvorgängen zu schützen, indem es Ihnen kontinuierliche Backups Ihrer Tabellendaten bietet.

9. Juli 2020

[Allgemeine Verfügbarkeit von Amazon Keyspaces](#)

Mit Amazon Keyspaces, das in der Vorversion als Amazon Managed Apache Cassandra Service (MCS) bekannt war, können Sie den Cassandra Query Language (CQL) -Code, Apache 2.0-lizenzierte Cassandra-Treiber und Entwicklertools verwenden, die Sie bereits heute verwenden.

23. April 2020

---

|                                                                                                |                                                                                                                                                                                                                                                                                        |                |
|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <a href="#">Automatische Skalierung von Amazon Keyspaces</a>                                   | Amazon Keyspaces (für Apache Cassandra) ist in Application Auto Scaling integriert, sodass Sie Durchsatzkapazität für variable Workloads als Reaktion auf den tatsächlichen Anwendungsverkehr effizient bereitstellen können, indem die Durchsatzkapazität automatisch angepasst wird. | 23. April 2020 |
| <a href="#">Schnittstelle für virtuelle private Cloud-Endpunkte (VPC) für Amazon Keyspaces</a> | Amazon Keyspaces bietet private Kommunikation zwischen dem Service und Ihrer VPC, sodass der Netzwerkverkehr das Amazon-Netzwerk nicht verlässt.                                                                                                                                       | 16. April 2020 |
| <a href="#">Tag-basierte Zugriffsrichtlinien</a>                                               | Sie können jetzt Ressourcen-Tags in IAM-Richtlinien verwenden, um den Zugriff auf Amazon Keyspaces zu verwalten.                                                                                                                                                                       | 8. April 2020  |
| <a href="#">Datentyp des Zählers</a>                                                           | Amazon Keyspaces hilft Ihnen jetzt dabei, Inkremente und Dekremente von Spaltenwerten mithilfe von Zählern zu koordinieren.                                                                                                                                                            | 7. April 2020  |
| <a href="#">Markieren von Ressourcen</a>                                                       | Amazon Keyspaces ermöglicht es Ihnen jetzt, Ressourcen mithilfe von Tags zu kennzeichnen und zu kategorisieren.                                                                                                                                                                        | 31. März 2020  |

[AWS CloudFormation Unterstützung](#)

Amazon Keyspaces hilft Ihnen jetzt dabei, die Erstellung und Verwaltung von Ressourcen zu automatisieren, indem Sie AWS CloudFormation.

25. März 2020

[Support für IAM-Rollen und -Richtlinien sowie SigV4-Authentifizierung](#)

Es wurden Informationen hinzugefügt, wie Sie [AWS Identity and Access Management \(IAM\)](#) zur Verwaltung von Zugriffsberechtigungen und zur Implementierung von Sicherheitsrichtlinien für Amazon Keyspaces verwenden können und wie Sie das Authentifizierungs-Plugin für den DataStax Java-Treiber für Cassandra verwenden können, um mithilfe von IAM-Rollen und föderierten Identitäten programmgesteuert auf Amazon Keyspaces zuzugreifen.

17. März 2020

[Kapazitätsmodus mit Lese-/Schreibzugriff](#)

Amazon Keyspaces unterstützt jetzt zwei Durchsatzkapazitätsmodi mit Lese-/Schreibzugriff. Der Lese-/Schreibkapazitätsmodus steuert, wie Ihnen der Lese- und Schreibdurchsatz in Rechnung gestellt wird und wie die Tabellendurchsatzkapazität verwaltet wird.

20. Februar 2020

## Erstversion

Diese Dokumentation behandelt die erste Version von Amazon Keyspaces (für Apache Cassandra).

3. Dezember 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.