aws

Entwicklerhandbuch

Amazon DocumentDB



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

Amazon DocumentDB: Entwicklerhandbuch

Copyright © 2025 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 DocumentDB ... 1
Übersicht ... 1
Cluster ... 3
Instances ... 4
Regionen und AZs ... 7
Regionen ... 7
Verfügbarkeitszonen ... 8
Preisgestaltung ... 10
Kostenlose Testversion ... 11
Überwachen ... 11
Schnittstellen ... 11
AWS Management Console ... 11
AWS CLI ... 11
MongoDB-Treiber ... 12
Als nächstes ... 12
Funktionsweise ... 12
Amazon DocumentDB DocumentDB-Endpunkte ... 15
TLS-Unterstützung ... 19
Amazon DocumentDB DocumentDB-Speicher ... 19
Amazon DocumentDB DocumentDB-Replikation ... 20
Zuverlässigkeit von Amazon DocumentDB ... 20
Lesen Sie die Einstellungsoptionen ... 22
TTL löscht ... 26
Abrechnungsfähige Ressourcen ... 27
Was ist eine Dokumentendatenbank? ... 30
Anwendungsfälle ... 31
Dokumente verstehen ... 32
Arbeiten mit Dokumenten ... 38
Leitfaden für die ersten Schritte ... 51
Voraussetzungen ... 52
Schritt 1: Erstellen eines Clusters ... 53
Schritt 2: Connect zu Ihrem Cluster her ... 56
Schritt 3: Daten einfügen und abfragen ... 58
Schritt 4: Erkunden ... 60

Schneller Start mit AWS CloudFormation 61
Voraussetzungen 62
Erforderliche IAM-Berechtigungen 62
EC2 Amazon-Schlüsselpaar 65
Einen Amazon DocumentDB DocumentDB-Stack AWS CloudFormation starten 65
Zugreifen auf den Amazon DocumentDB-Cluster 71
Kündigungsschutz und Löschschutz 71
MongoDB-Kompatibilität 72
MongoDB 5.0-Kompatibilität 72
Was ist neu in Amazon DocumentDB 5.0 72
Erste Schritte mit Amazon DocumentDB 5.0 73
Upgrade oder Migration auf Amazon DocumentDB 5.0 74
Funktionelle Unterschiede 74
MongoDB 4.0-Kompatibilität 76
Funktionen von Amazon DocumentDB 4.0 76
Erste Schritte mit Amazon DocumentDB 4.0 77
Upgrade oder Migration zu Amazon DocumentDB 4.0 78
Funktionelle Unterschiede 78
Transaktionen 80
Voraussetzungen 80
Bewährte Methoden 81
Einschränkungen 81
Überwachung und Diagnose 82
Isolationsstufe für Transaktionen 83
Anwendungsfälle 83
Transaktionen mit mehreren Kontoauszügen 84
Transaktionen mit mehreren Sammelaktionen 85
Beispiele für Transaktions-APIs für die Callback-API 87
Beispiele für Transaktions-APIs für die Kern-API 87
Unterstützte Befehle 121
Nicht unterstützte Funktionen 122
Sitzungen 123
Kausale Konsistenz 123
Wiederholbare Schreibvorgänge 124
Transaktionsfehler 125
Bewährte Methoden 126

Grundlegende Anleitungen für den Betrieb 126
Dimensionierung der Instanzen 128
Arbeiten mit Indizes 129
Erstellen von Indizes 129
Index-Selektivität 130
Einfluss von Indizes auf das Schreiben von Daten 130
Identifizierung fehlender Indizes 131
Identifizieren ungenutzter Indizes 131
Bewährte Methoden für die Gewährleistung der Sicherheit 131
Kostenoptimierung 132
Verwendung von Metriken zur Identifizierung von Problemen mit der Leistung 133
Anzeigen von Leistungsmetriken 133
Einen CloudWatch Alarm einrichten 134
Auswerten von Leistungsmetriken 134
Auswertung der Nutzung von Amazon DocumentDB DocumentDB-Instances anhand von Metriken CloudWatch 136
Optimieren von Abfragen 137
TTL- und Zeitreihen-Workloads 138
Migrationen 138
Arbeiten mit Cluster-Parametergruppen 139
Abfragen in der Aggregationspipeline 139
`batchInsert` und `batchUpdate` 139
Funktionale Unterschiede zu MongoDB 140
Funktionale Vorteile von Amazon DocumentDB 140
Implizite Transaktionen 140
Funktionelle Unterschiede wurden aktualisiert 141
Array-Indizierung 142
Indizes mit mehreren Schlüsseln 143
Nullzeichen in Zeichenketten 144
Rollenbasierte Zugriffskontrolle 144
`$regex`Indizierung 144
Projektion für verschachtelte Dokumente 145
Funktionale Unterschiede zu MongoDB 145
`$vectorSearch`-Operator 146
`OpCountersCommand` 146
Admin-Datenbanken und Sammlungen 146

cursormaxTimeMS ... 146
explain() ... 147
Der Index wird erstellt ... 147
Suche mit leerem Schlüssel im Pfad ... 147
MongoDB APIs, Operationen und Datentypen ... 148
mongodumpund mongorestore Dienstprogramme ... 148
Reihenfolge der Ergebnisse ... 148
Wiederholbare Schreibvorgänge ... 149
Sparse-Index ... 149
$elemMatchVerwendung innerhalb eines Ausdrucks $all ... 150
$ne,,$nin, nornot$exists, und $elemMatch Indizierung ... 150
Dollar ($) und Punkt (.) in Feldnamen ... 151
$lookup ... 151
$naturalund umgekehrte Sortierung ... 155
Unterstützte MongoDB APIs, Operationen und Datentypen ... 156
Datenbankbefehle ... 157
Administrative Befehle ... 157
Aggregation ... 159
Authentifizierung ... 159
Diagnosebefehle ... 159
Abfrage- und Schreiboperationen ... 160
Befehle zur Rollenverwaltung ... 161
Befehle für Sitzungen ... 162
Benutzerverwaltung ... 163
Sharding-Befehle ... 163
Abfrage- und Projektionsoperatoren ... 165
Array-Operatoren ... 166
Bitweise Operatoren ... 166
Operator kommentieren ... 166
Vergleichsoperatoren ... 167
Elementoperatoren ... 167
Operatoren für Bewertungsabfragen ... 167
Logische Operatoren ... 168
Projektionsoperatoren ... 168
Operatoren aktualisieren ... 169
Array-Operatoren ... 169

Bitweise Operatoren ... 170
Feldoperatoren ... 170
Modifikatoren aktualisieren ... 170
Geodaten ... 171
Geometrie-Spezifizierer ... 171
Selektoren abfragen ... 172
Cursor-Methoden ... 172
Operatoren für Aggregationspipelines ... 174
Akkumulatorausdrücke ... 175
Arithmetische Operatoren ... 176
Array-Operatoren ... 177
Boolesche Operatoren ... 178
Vergleichsoperatoren ... 179
Operatoren für bedingte Ausdrücke ... 179
Datentyp-Operator ... 179
Operator zur Datengröße ... 180
Datumsoperatoren ... 180
Wörtlicher Operator ... 181
Operator zusammenführen ... 181
Natürlicher Operator ... 182
Satzoperatoren ... 182
Bühnenbetreiber ... 183
Zeichenfolgenoperatoren ... 184
Systemvariablen ... 186
Operator für die Textsuche ... 186
Geben Sie Konvertierungsoperatoren ein ... 186
Operatoren für Variablen ... 187
Verschiedene Betreiber ... 187
Datentypen ... 188
Indizes und Indexeigenschaften ... 189
Indizes ... 189
Eigenschaften indizieren ... 190
Generative KI ... 191
SageMaker Leinwand ... 191
Wie erstellt man ML-Modelle ohne Code mit AI Canvas SageMaker ... 191
Konfiguration der AI-Domäne und des Benutzerprofils SageMaker ... 192

Konfiguration von IAM-Zugriffsberechtigungen für Amazon DocumentDB und AI Canvas SageMaker ... 192
Datenbankbenutzer und Rollen für SageMaker AI Canvas erstellen ... 193
Verfügbare Regionen ... 193
Vektor-Suche ... 194
Vektoren einfügen ... 195
Einen Vektorindex erstellen ... 195
Eine Indexdefinition abrufen ... 199
Vektoren abfragen ... 200
Funktionen und Einschränkungen ... 204
Bewährte Methoden ... 206
Migration zu Amazon DocumentDB ... 207
Migrationstools ... 207
AWS Database Migration Service ... 207
Befehlszeilen-Dienstprogramme ... 208
Erkennung ... 208
Planung: Amazon DocumentDB-Clusteranforderungen ... 212
Migrationsansätze ... 215
Offline ... 216
Status "Online" ... 217
Hybrid ... 219
Quellen für die Migration ... 220
Konnektivität bei der Migration ... 221
Testen ... 224
Überlegungen zum Testen des Migrationsplans ... 225
Leistungstests ... 228
Failover-Tests ... 229
Weitere Ressourcen ... 229
Leitfaden für die Migration ... 229
Migrationsprozess ... 229
Weitere Ressourcen ... 234
Aktualisierung der Amazon DocumentDB DocumentDB-Engine-Version ... 236
MVU-Voraussetzungen und Einschränkungen ... 237
Bewährte Methoden für direkte Upgrades von Hauptversionen ... 238
Testen Sie direkte Hauptversions-Upgrades mit geklonten Clustern ... 239
Vor einem direkten Upgrade der Hauptversion ... 239

Während eines direkten Hauptversions-Upgrades ... 241
Nach einem direkten Upgrade der Hauptversion ... 242
Durchführung eines direkten Hauptversions-Upgrades ... 244
Unterschiede zwischen auf Amazon DocumentDB 3.6/4.0 auf 5.0 aktualisierten Clustern und neuen Amazon DocumentDB 5.0-Clustern ... 247
Fehlerbehebung bei einem direkten Upgrade der Hauptversion ... 248
Ein Upgrade durchführen mit AWS DMS ... 249
Schritt 1: Change Streams aktivieren ... 249
Schritt 2: Ändern Sie die Aufbewahrungsdauer für Change-Streams ... 250
Schritt 3: Migrieren Sie Ihre Indizes ... 250
Schritt 4: Erstellen Sie eine AWS DMS Replikationsinstanz ... 252
Schritt 5: Erstellen Sie einen AWS DMS Quellendpunkt ... 254
Schritt 6: Erstellen Sie einen AWS DMS Zielendpunkt ... 256
Schritt 7: Erstellen Sie eine Migrationsaufgabe und führen Sie sie aus ... 259
Schritt 8: Ändern des Anwendungsendpunkts auf den Amazon DocumentDB DocumentDB-Zielcluster ... 261
Sicherheit ... 262
Passwortverwaltung mit AWS Secrets Manager ... 263
Einschränkungen für die Secrets Manager Manager-Integration mit Amazon DocumentDB . 264
Überblick über die Verwaltung von Primärbenutzerkennwörtern mit AWS Secrets Manager ... 264
Durchsetzung der Amazon DocumentDB DocumentDB-Verwaltung des primären Benutzerkennworts in AWS Secrets Manager ... 265
Verwaltung des primären Benutzerpassworts für einen Cluster mit Secrets Manager ... 265
Datenschutz ... 266
Clientseitige Verschlüsselung auf Feldebene ... 267
Verschlüsseln von Daten im Ruhezustand ... 275
Verschlüsseln von Daten während der Übertragung ... 281
Schlüsselverwaltung ... 292
Identitäts- und Zugriffsverwaltung ... 292
Zielgruppe ... 293
Authentifizierung mit Identitäten ... 294
Verwalten des Zugriffs mit Richtlinien ... 298
So funktioniert Amazon DocumentDB mit IAM ... 301
Beispiele für identitätsbasierte Richtlinien ... 309
Fehlerbehebung ... 313

Verwaltung der Zugriffsberechtigungen für Ihre Amazon DocumentDB DocumentDB-Ressourcen ... 315
Verwenden von identitätsbasierten Richtlinien (IAM-Richtlinien) ... 320
AWS verwaltete Richtlinien für Amazon DocumentDB ... 325
Referenz zu Amazon DocumentDB DocumentDB-API-Berechtigungen ... 344
Authentifizierung mit IAM-Identität ... 353
Erste Schritte mit IAM-Benutzern und -Rollen ... 354
Compute-Typen konfigurieren ... 358
Überwachung von Authentifizierungsanfragen ... 358
Verwenden der IAM-Authentifizierung ... 359
Treiber, die IAM unterstützen ... 359
Häufig gestellte Fragen zur IAM-Identitätsauthentifizierung ... 360
Amazon DocumentDB DocumentDB-Benutzer verwalten ... 360
Primär und Benutzer `serviceadmin` ... 361
Zusätzliche Benutzer erstellen ... 362
Automatisches Rotieren von Passwörtern ... 364
Rollenbasierte Zugriffssteuerung ... 364
RBAC-Konzepte ... 365
Erste Schritte mit den integrierten RBAC-Rollen ... 368
Erste Schritte mit benutzerdefinierten RBAC-Rollen ... 371
Als Benutzer eine Verbindung zu Amazon DocumentDB herstellen ... 376
Allgemeine Befehle ... 378
Funktionelle Unterschiede ... 383
Einschränkungen ... 383
Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle ... 383
Protokollierung und Überwachung ... 393
Aktualisierung von Zertifikaten ... 393
Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters ... 394
Automatische Rotation von Serverzertifikaten ... 399
Fehlerbehebung ... 400
Häufig gestellte Fragen ... 400
Aktualisierung von Zertifikaten — GovCloud ... 407
Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters ... 394
Fehlerbehebung ... 400
Häufig gestellte Fragen ... 400
Compliance-Validierung ... 421

Ausfallsicherheit ... 422
Sicherheit der Infrastruktur ... 423
VPC-Endpunkte (AWS PrivateLink) ... 424
Überlegungen zu VPC-Endpunkten ... 425
Verfügbarkeit in Regionen ... 425
Erstellen eines VPC-Schnittstellen-Endpunkts für die Amazon DocumentDB DocumentDB-API ... 426
Erstellen einer VPC-Endpunktrichtlinie für die Amazon DocumentDB DocumentDB-API ... 427
Bewährte Methoden für die Gewährleistung der Sicherheit ... 428
Prüfereignisse ... 429
Unterstützte Ereignisse ... 430
Auditing aktivieren ... 436
Deaktivieren Sie die Überwachung ... 443
Greifen Sie auf Ihre Ereignisse zu ... 446
DML-Ereignisse filtern ... 446
Sichern und Wiederherstellen ... 453
Sichern und Wiederherstellen: Konzepte ... 454
Grundlegendes zur Sicherungsspeicher-Nutzung in ... 457
Daten speichern, wiederherstellen, importieren und exportieren ... 459
`mongodump` ... 459
`mongorestore` ... 460
`mongoexport` ... 460
`mongoimport` ... 461
Tutorial ... 462
Überlegungen zu Cluster-Snapshots ... 464
Backup-Speicher ... 465
Backup-Fenster ... 466
Aufbewahrungszeitraum für Backups ... 467
Kopieren Sie die Cluster-Snapshot- ... 468
Vergleich von automatischen und manuellen Snapshots ... 468
Einen manuellen Cluster-Snapshot erstellen ... 471
Einen Cluster-Snapshot kopieren ... 474
Kopieren freigegebener Snapshots ... 475
Schnappschüsse zwischen AWS-Regionen ... 475
Einschränkungen ... 475
Umgang mit Verschlüsselungen ... 476

Überlegungen zu Parametergruppen ... 476
Einen Cluster-Snapshot kopieren ... 477
Freigeben eines Cluster-Snapshots ... 484
Freigeben eines verschlüsselten Snapshots ... 485
Freigeben eines Snapshots ... 488
Wiederherstellung aus einem Cluster-Snapshot ... 492
Zu einem bestimmten Zeitpunkt wiederherstellen ... 500
Löschen eines Cluster-Snapshots ... 506
Amazon DocumentDB verwalten ... 509
Überblick über die operativen Aufgaben ... 509
Hinzufügen eines Replikats zu einem Amazon DocumentDB-Cluster ... 510
Beschreibung von Clustern und Instanzen ... 511
Einen Cluster-Snapshot erstellen ... 513
Wiederherstellung aus einem Snapshot ... 514
Eine Instanz aus einem Cluster entfernen ... 515
Löschen eines Clusters ... 516
Globale Cluster ... 517
Was ist ein globaler Cluster? ... 517
Wie sind globale Cluster nützlich? ... 517
Was sind die aktuellen Einschränkungen globaler Cluster? ... 517
Schnellstart-Anleitung ... 518
Verwaltung globaler Cluster ... 535
Globale Cluster verbinden ... 543
Überwachung globaler Cluster ... 544
Notfallwiederherstellung ... 545
Verwalten von -Clustern ... 562
Cluster verstehen ... 562
Cluster-Einstellungen ... 565
Cluster-Speicherkonfigurationen ... 568
Den Status eines Clusters ermitteln ... 571
Cluster-Lebenszyklus ... 573
Skalierung von Clustern ... 617
Klonen eines Volumes für einen Cluster ... 621
Die Cluster-Fehlertoleranz verstehen ... 635
Verwalten von Instances ... 636
Den Status einer Instance ermitteln ... 637

Instance-Lebenszyklus ... 637
Instanzklassen verwalten ... 662
NVMe-gestützte Instances ... 673
Verwaltung von Subnetzgruppen ... 676
Erstellen einer Subnetzgruppe ... 677
Beschreibung einer Subnetzgruppe ... 683
Ändern einer Subnetzgruppe ... 686
Löschen einer Subnetzgruppe ... 689
Hohe Verfügbarkeit und Replikation ... 691
Skalierung von Lesevorgängen ... 691
Hohe Verfügbarkeit ... 691
Hinzufügen von -Replicas ... 693
Failover ... 694
Replikationsverzögerung ... 698
Verwalten von Indexen ... 700
Amazon DocumentDB DocumentDB-Indexerstellung ... 700
Pflege Amazon DocumentDB DocumentDB-Indizes ... 706
Verwaltung der Dokumentenkomprimierung ... 707
Verwaltung der Dokumentenkomprimierung ... 708
Überwachung der Dokumentenkomprimierung ... 710
Verwaltung von Ereignissen ... 711
Veranstaltungskategorien anzeigen ... 712
Amazon DocumentDB DocumentDB-Ereignisse anzeigen ... 714
Auswahl von Regionen und Availability Zones ... 717
Verfügbarkeit in Regionen ... 718
Verwaltung von Cluster-Parametergruppen ... 720
Beschreibung von Cluster-Parametergruppen ... 721
Cluster-Parametergruppen erstellen ... 728
Cluster-Parametergruppen ändern ... 731
Modifizieren von Clustern zur Verwendung benutzerdefinierter Cluster-Parametergruppen ... 736
Cluster-Parametergruppen werden kopiert ... 738
Cluster-Parametergruppen werden zurückgesetzt ... 740
Cluster-Parametergruppen löschen ... 743
Referenz zu den Cluster-Parametern ... 746
Endpunkte verstehen ... 765
Die Endpunkte eines Clusters finden ... 766

Den Endpunkt einer Instanz finden ... 768
Verbindung zu Endpunkten herstellen ... 772
Amazon DocumentDB verstehen ARNs ... 773
Konstruieren eines ARN ... 774
Suchen eines ARNs ... 777
Taggen von -Ressourcen ... 779
Übersicht über -Ressourcen-Tags ... 780
Tag-Einschränkungen ... 781
Hinzufügen oder Aktualisieren von Tags ... 781
Auflisten von Tags ... 783
Entfernen von Tags ... 785
Wartung von Amazon DocumentDB ... 786
Engine-Patch-Benachrichtigungen ... 787
Ausstehende Wartungsarbeiten anzeigen ... 789
Engine-Updates ... 791
Vom Benutzer initiierte Updates ... 794
Wartungsfenster verwalten ... 796
Betriebssystemupdates ... 798
Grundlegendes zu Rollen im Zusammenhang mit Services ... 801
Berechtigungen von serviceverknüpften Rollen ... 802
Erstellen einer serviceverknüpften Rolle ... 804
Änderung einer serviceverknüpften Rolle ... 804
Löschen einer serviceverknüpften Rolle ... 805
Unterstützte Regionen für serviceverknüpfte Amazon DocumentDB DocumentDB-Rollen ... 806
Change-Streams verwenden ... 806
Unterstützte Vorgänge ... 807
Fakturierung ... 807
Einschränkungen ... 807
Change-Streams aktivieren ... 809
Change-Streams mit Python verwenden ... 810
Vollständige Suche nach Dokumenten ... 813
Wiederaufnahme eines Change-Streams ... 814
Wiederaufnahme mit `startAtOperationTime` ... 816
Wiederaufnahme mit `postBatchResumeToken` ... 818
Transaktionen ... 819
Änderung der Aufbewahrungsdauer von Protokollen ... 819

Streams auf sekundären Instanzen ändern ... 823
Verwendung AWS Lambda mit Change-Streams ... 824
Einschränkungen ... 825
Verwendung elastischer Cluster ... 826
Anwendungsfälle für elastische Cluster ... 826
Benutzerprofile ... 827
Inhaltsverwaltung und historische Aufzeichnungen ... 827
Vorteile elastischer Cluster ... 827
AWS Serviceintegration ... 827
Verfügbarkeit von Regionen und Versionen ... 828
Regionale Verfügbarkeit für elastische Cluster ... 828
Verfügbarkeit von Versionen ... 829
Einschränkungen ... 829
Elastische Clusterverwaltung ... 829
Abfrage- und Schreiboperationen ... 830
Sammlungs- und Indexverwaltung ... 830
Verwaltung und Diagnose ... 830
Opt-in-Funktionen ... 831
Funktionsweise ... 831
Elastisches Cluster-Sharding von Amazon DocumentDB ... 831
Elastische Cluster-Migration ... 835
Elastische Cluster-Skalierung ... 835
Elastische Cluster-Zuverlässigkeit ... 835
Elastischer Cluster-Speicher und Verfügbarkeit ... 835
Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und Elastic Clusters ... 836
Erste Schritte ... 837
Voraussetzungen ... 838
Schritt 1: Erstellen Sie einen elastischen Cluster ... 839
Schritt 2: Connect zu Ihrem Elastic Cluster her ... 846
Schritt 3: Teilen Sie Ihre Sammlung, fügen Sie Daten ein und fragen Sie sie ab ... 847
Schritt 4: Erkunden ... 850
Bewährte Methoden ... 851
Auswahl von Shard-Schlüsseln ... 851
Verbindungsverwaltung ... 852
Sammlungen, die nicht geteilt wurden ... 852
Skalierung elastischer Cluster ... 852

Überwachung elastischer Cluster ... 853
Verwaltung elastischer Cluster ... 853
Modifizieren elastischer Cluster-Konfigurationen ... 854
Überwachen eines elastischen Clusters ... 857
Löschen eines elastischen Clusters ... 861
Verwaltung von Elastic Cluster-Snapshots ... 863
Stoppen und Starten eines elastischen Clusters ... 879
Aufrechterhaltung elastischer Cluster ... 883
Datenverschlüsselung im Ruhezustand ... 892
So verwenden elastische Amazon DocumentDB-Cluster Zuschüsse in AWS KMS ... 894
Einen kundenverwalteten Schlüssel erstellen ... 895
Überwachung Ihrer Verschlüsselungsschlüssel für Amazon DocumentDB Elastic Clusters ... 896
Weitere Informationen ... 902
Service-verknüpfte Rollen ... 902
Dienstbezogene Rollenberechtigungen für elastische Cluster ... 903
Überwachung von Amazon DocumentDB ... 907
Den Status eines Clusters überwachen ... 909
Werte für den Cluster-Status ... 909
Den Status eines Clusters überwachen ... 911
Den Status einer Instance überwachen ... 913
Instance-Statuswerte ... 913
Überwachen Sie den Instance-Status mit dem AWS Management Console oder AWS CLI ... 916
Werte für den Integritätsstatus der Instanz ... 918
Überwachen des Zustands der Instanz mit dem AWS Management Console ... 918
Amazon DocumentDB DocumentDB-Empfehlungen anzeigen ... 920
Ereignisabonnements ... 923
Abonnieren von Ereignissen ... 924
-Abonnements verwalten ... 927
Kategorien und Nachrichten ... 931
Überwachen von Amazon DocumentDB mit CloudWatch ... 935
Amazon DocumentDB-Metriken ... 935
CloudWatch Daten anzeigen ... 953
Abmessungen von Amazon DocumentDB ... 959
Überwachung von Opcounter-Metriken ... 960
Überwachung von Datenbankverbindungen ... 960
Protokollieren Amazon DocumentDB DocumentDB-API-Aufrufen mit CloudTrail ... 960

Amazon DocumentDB DocumentDB-Informationen in CloudTrail 961
Profilierung von Vorgängen 962
Unterstützte Vorgänge 963
Einschränkungen 963
Den Profiler aktivieren 963
Den Profiler deaktivieren 968
Export von Profiler-Protokollen deaktivieren 969
Greifen Sie auf Ihre Profiler-Protokolle zu 972
Häufig gestellte Fragen 972
Überwachung mit Performance Insights 973
Konzepte von Performance Insights 974
Aktivieren und Deaktivieren von Performance Insights 978
Konfigurieren von Zugriffsrichtlinien für Performance Insights 981
Analyse der Metriken mit dem Performance Insights-Dashboard 987
Abrufen von Metriken mit der Performance Insights-API 1007
CloudWatch Amazon-Metriken für Performance Insights 1022
Performance Insights für Zählermetriken 1024
OpenSearch Integration 1027
Amazon OpenSearch Service als Ziel 1027
Schritt 1: Amazon OpenSearch Service-Domain oder OpenSearch serverlose Sammlung erstellen 1028
Schritt 2: Change-Streams im Amazon DocumentDB-Cluster aktivieren 1028
Schritt 3: Richten Sie die Pipeline-Rolle mit Schreibberechtigungen für den Amazon S3 S3-Bucket und die Zieldomäne oder Sammlung ein 1028
Schritt 4: Fügen Sie die für die Pipeline-Rolle erforderlichen Berechtigungen hinzu, um X-ENI zu erstellen 1029
Schritt 5: Erstellen Sie die Pipeline 1030
Einschränkungen 1030
Entwickeln mit Amazon DocumentDB 1032
Mit DocumentDB Connect 1032
Programmgesteuert eine Verbindung herstellen 1032
Verbindung von außerhalb einer Amazon VPC herstellen 1061
Als Replikatsatz eine Verbindung herstellen 1063
Connect mit Studio 3T her 1068
Connect mit DataGrip 1079
Stellen Sie über Amazon eine Verbindung her EC2 1087

Stellen Sie mithilfe des JDBC-Treibers eine Verbindung her ... 1128
Stellen Sie mithilfe des ODBC-Treibers eine Verbindung her ... 1146
Programm mit DocumentDB ... 1162
DocumentDB Java-Programmierhandbuch ... 1162
Verwenden der JSON-Schemavalidierung ... 1196
Kontingente und -Einschränkungen ... 1207
Unterstützte Instance-Typen ... 1207
Unterstützte -Regionen ... 1209
Regionale Kontingente ... 1210
Aggregationsgrenzen ... 1213
Cluster-Grenzwerte ... 1213
Instance-Limits ... 1215
Benennungseinschränkungen: ... 1217
TTL-Einschränkungen ... 1219
Elastische Cluster-Grenzwerte ... 1219
Limits für elastische Cluster-Shards ... 1220
CPU-, Speicher-, Verbindungs- und Cursorlimits für Elastic Cluster pro Shard ... 1221
Abfragen ... 1222
Dokumente abfragen ... 1222
Alle Dokumente werden abgerufen ... 1223
Passende Feldwerte ... 1223
Eingebettete Dokumente ... 1223
Feldwerte in eingebetteten Dokumenten ... 1224
Abgleich mit einem Array ... 1224
Übereinstimmende Werte in einem Array ... 1224
Verwenden von Operatoren ... 1225
Abfrageplan ... 1225
Abfrageplan ... 1225
Plan-Cache abfragen ... 1227
Erläutern Sie die Ergebnisse ... 1227
Phase des Scannens und Filterns ... 1228
Index-Kreuzung ... 1229
Indexunion ... 1230
Schnittung/Vereinigung mehrerer Indizes ... 1231
Zusammengesetzter Index ... 1232
Phase sortieren ... 1232

Gruppenphase ... 1232
Koordinatenbasierte Daten ... 1233
Übersicht ... 1
Indizierung und Speicherung von Geodaten ... 1233
Abfragen von koordinatenbasierten Daten ... 1235
Einschränkungen ... 1239
Teilweiser Index ... 1239
Erstellen Sie einen Teilindex ... 1239
Unterstützte Operatoren ... 1240
Abfrage mit einem Teilindex ... 1240
Funktionen des partiellen Indexes ... 1241
Einschränkungen des partiellen Indexes ... 1245
Textsuche ... 1246
Unterstützte Funktionen ... 1247
Verwenden des Amazon DocumentDB-Textindex ... 1247
Unterschiede zu MongoDB ... 1253
Bewährte Verfahren und Richtlinien ... 1253
Einschränkungen ... 1254
Fehlerbehebung ... 1255
Verbindungsprobleme ... 1255
Es kann keine Verbindung zu einem Amazon DocumentDB DocumentDB-Endpunkt hergestellt werden ... 1255
Testen einer Verbindung zu einer Amazon DocumentDB DocumentDB-Instance ... 1261
Verbindung zu einem ungültigen Endpunkt herstellen ... 1261
Die Treiberkonfiguration wirkt sich auf die Anzahl der Verbindungen aus ... 1262
Indexerstellung ... 1262
Die Indexerstellung schlägt fehl ... 1263
Latenzprobleme und Fehler bei der Indexerstellung im Hintergrund ... 1263
Leistung und Ressourcennutzung ... 1264
Statistiken zum Einfügen, Aktualisieren und Löschen anzeigen ... 1265
Analysieren Sie die Cache-Leistung ... 1266
Finden und Beenden von langsamen und blockierten Abfragen? ... 1268
Anzeigen eines Abfrageplans und Optimieren einer Abfrage ... 1269
Wie kann ich mir einen Abfrageplan in elastischen Clustern ansehen? ... 1271
Auflisten aller aktuell auf einer Instance ausgeführten Operationen ... 1274
Ermitteln, wann eine Abfrage Fortschritte macht ... 1276

Ermitteln, warum ein System plötzlich langsam ausgeführt wird ... 1279
Ermitteln der Ursache einer hohen CPU-Nutzung ... 1281
Finde die offenen Cursor auf einer Instanz ... 1282
Aktuelle Version der Amazon DocumentDB DocumentDB-Engine anzeigen ... 1282
Analysieren Sie die Indexnutzung und identifizieren Sie ungenutzte Indizes ... 1283
Identifizieren Sie fehlende Indizes ... 1285
Zusammenfassung nützlicher Abfragen ... 1287
API-Referenz zur Ressourcenverwaltung ... 1289
Aktionen ... 1289
Amazon DocumentDB (with MongoDB compatibility) ... 1292
Amazon DocumentDB Elastic Clusters ... 1486
Datentypen ... 1562
Amazon DocumentDB (with MongoDB compatibility) ... 1564
Amazon DocumentDB Elastic Clusters ... 1645
Häufige Fehler ... 1664
Geläufige Parameter ... 1665
Versionshinweise ... 1669
18. Juni 2025 ... 1671
Neues Feature ... 1671
8. Mai 2025 ... 1672
Neues Feature ... 1672
2. April 2025 ... 1672
Neues Feature ... 1672
Fehlerkorrekturen und andere Änderungen ... 1672
24. März 2025 ... 1673
Neues Feature ... 1673
6. Februar 2025 ... 1673
Neues Feature ... 1673
28. Januar 2025 ... 1673
Neues Feature ... 1673
15. Januar 2025 ... 1673
Neue Features ... 1673
Fehlerkorrekturen und andere Änderungen ... 1674
18. Dezember 2024 ... 1674
Neue Features ... 1674
12. November 2024 ... 1674

Neue Features 1674
6. November 2024 1675
Neue Features 1675
1. November 2024 1675
Neues Feature 1675
22. Oktober 2024 1675
Neues Feature 1675
18. September 2024 1676
Neues Feature 1676
17. September 2024 1676
Neue Features 1676
Fehlerkorrekturen und andere Änderungen 1676
22. August 2024 1677
Neues Feature 1677
20. August 2024 1677
Neues Feature 1677
8. August 2024 1677
Neues Feature 1677
23. Juli 2024 1678
Neue Features 1678
Fehlerkorrekturen und andere Änderungen 1679
22. Juli 2024 1679
Neue Features 1679
Fehlerkorrekturen und andere Änderungen 1680
9. Juli 2024 1680
Neues Feature 1680
8. Juli 2024 1680
Neues Feature 1680
25. Juni 2024 1680
Neues Feature 1680
29. Mai 2024 1681
Neue Features 1681
3. April 2024 1681
Neue Features 1681
Fehlerkorrekturen und andere Änderungen 1682
22. Februar 2024 1682

Neue Features 1682
30. Januar 2024 1683
Neue Features 1683
10. Januar 2024 1683
Neue Features 1683
Fehlerkorrekturen und andere Änderungen 1684
20. Dezember 2023 1685
Weitere Änderungen 1685
13. Dezember 2023 1685
Neue Features 1685
29. November 2023 1685
Neue Features 1685
21. November 2023 1685
Neue Features 1685
17. November 2023 1686
Neue Features 1686
Fehlerkorrekturen und andere Änderungen 1686
6. November 2023 1686
Neue Features 1686
Fehlerkorrekturen und andere Änderungen 1686
25. September 2023 1687
Neue Features 1687
20. September 2023 1687
Neue Features 1687
15. September 2023 1687
Neue Features 1687
11. September 2023 1687
Neue Features 1687
3. August 2023 1687
Neue Features 1687
13. Juli 2023 1688
Neue Features 1688
Fehlerkorrekturen und andere Änderungen 1688
7. Juni 2023 1689
Fehlerkorrekturen und andere Änderungen 1689
10. Mai 2023 1689

Fehlerkorrekturen und andere Änderungen 1689
4. April 2023 1689
Fehlerkorrekturen und andere Änderungen 1689
22. März 2023 1690
Neue Features 1690
1. März 2023 1690
Neue Features 1690
27. Februar 2023 1691
Fehlerkorrekturen und andere Änderungen 1691
2. Februar 2023 1691
Fehlerkorrekturen und andere Änderungen 1691
30. November 2022 1692
Neue Features 1692
09. August 2022 1692
Neue Features 1692
Fehlerkorrekturen und andere Änderungen 1692
25. Juli 2022 1693
Neue Features 1693
27. Juni 2022 1693
Neue Features 1693
29. April 2022 1693
Neue Features 1693
7. April 2022 1693
Neue Features 1693
16. März 2022 1694
Neue Features 1694
8. Februar 2022 1694
Neue Features 1694
24. Januar 2022 1694
Neue Features 1694
21. Januar 2022 1695
Neue Features 1695
25. Oktober 2021 1695
Neue Features 1695
Fehlerkorrekturen und andere Änderungen 1696
24. Juni 2021 1696

Neue Features 1696
4. Mai 2021 1696
Neue Features 1696
Fehlerkorrekturen und andere Änderungen 1697
15. Januar 2021 1697
Neue Features 1697
9. November 2020 1698
Neue Features 1698
Fehlerkorrekturen und andere Änderungen 1699
30. Oktober 2020 1700
Neue Features 1700
Fehlerkorrekturen und andere Änderungen 1700
22. September 2020 1701
Neue Features 1701
Fehlerkorrekturen und andere Änderungen 1701
10. Juli 2020 1701
Neue Features 1701
Fehlerkorrekturen und andere Änderungen 1702
30. Juni 2020 1702
Neue Features 1702
Fehlerkorrekturen und andere Änderungen 1702
Dokumentverlauf 1703
..... mdccxvi

Was ist Amazon DocumentDB (mit MongoDB-Kompatibilität)

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein schneller, zuverlässiger und vollständig verwalteter Datenbankservice. Amazon DocumentDB macht es einfach, MongoDB-kompatible Datenbanken in der Cloud einzurichten, zu betreiben und zu skalieren. Mit Amazon DocumentDB können Sie denselben Anwendungscode ausführen und dieselben Treiber und Tools verwenden, die Sie mit MongoDB verwenden.

Bevor Sie Amazon DocumentDB verwenden, sollten Sie sich mit den unter beschriebenen Konzepten und Funktionen vertraut machen. Funktionsweise Anschließend führen Sie die Schritte unter Leitfaden für die ersten Schritte aus.

Themen

- Überblick über Amazon DocumentDB
- Cluster
- Instances
- Regionen und Availability Zones
- Amazon DocumentDB — Preise
- Überwachen
- Schnittstellen
- Als nächstes
- Amazon DocumentDB: So funktioniert's
- Was ist eine Dokumentendatenbank?

Überblick über Amazon DocumentDB

Im Folgenden sind einige wichtige Funktionen von Amazon DocumentDB aufgeführt:

- Amazon DocumentDB unterstützt zwei Arten von Clustern: instanzbasierte Cluster und elastische Cluster. Elastic Cluster unterstützen Workloads mit einer Speicherkapazität von Millionen reads/writes pro Sekunde und Petabyte. Weitere Informationen zu elastischen Clustern finden Sie unter. Verwendung elastischer Amazon DocumentDB-Cluster Der folgende Inhalt bezieht sich auf Amazon DocumentDB DocumentDB-Instance-basierte Cluster.

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

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

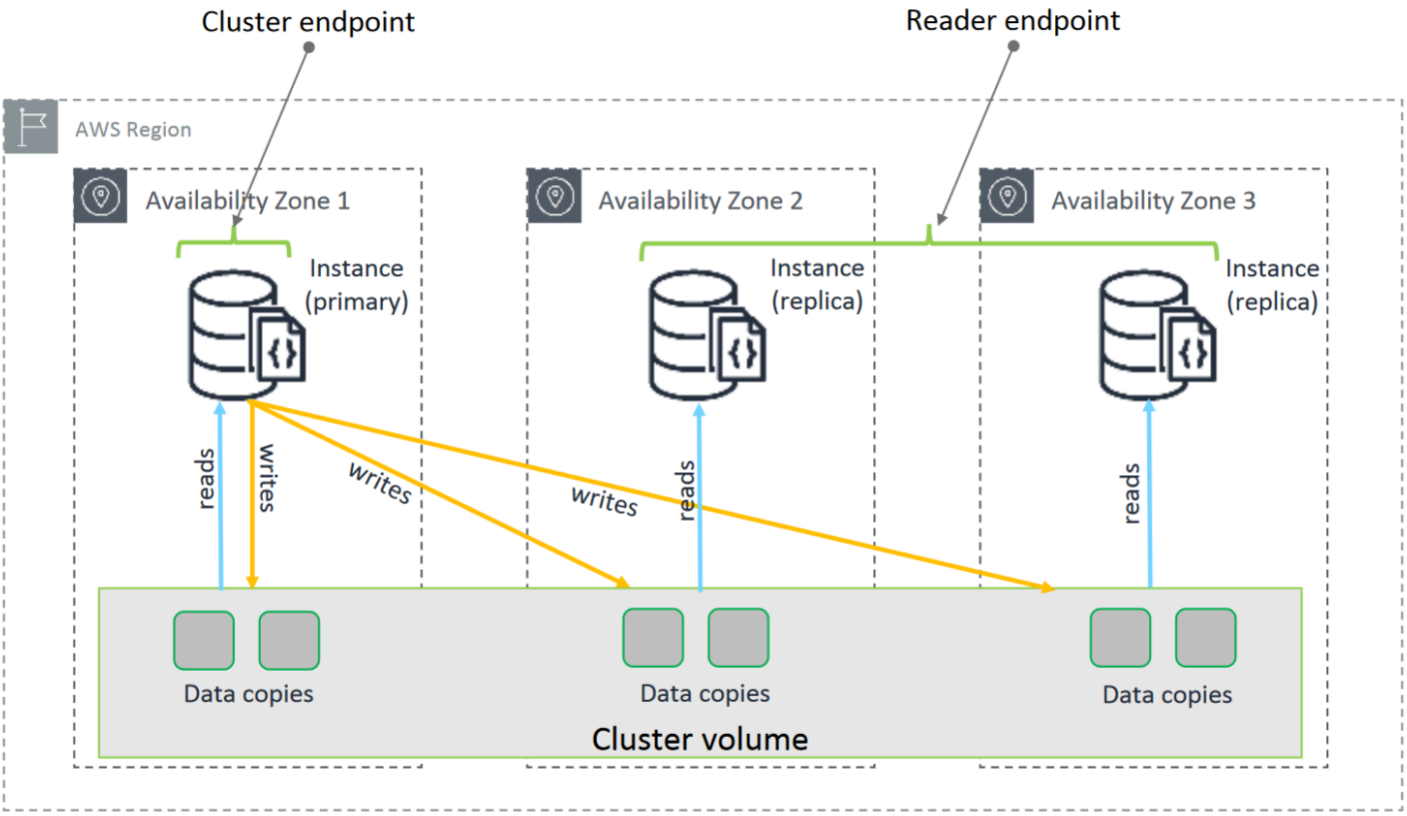
- Mit Amazon DocumentDB können Sie Ihre Datenbanken mit Schlüsseln verschlüsseln, die Sie über AWS Key Management Service ()AWS KMS erstellen und kontrollieren. In einem Datenbank-Cluster, der mit Amazon DocumentDB DocumentDB-Verschlüsselung ausgeführt wird, werden Daten, die im Ruhezustand im zugrunde liegenden Speicher gespeichert sind, verschlüsselt. Die automatischen Sicherungen, Snapshots und Replicas im gleichen Cluster werden ebenfalls verschlüsselt.
- Amazon DocumentDB ist im Rahmen des Federal Risk and Authorization Management Program (FedRAMP) autorisiert. Es verfügt über eine FedRAMP High-Autorisierung für AWS GovCloud (US-) Regionen und eine FedRAMP Moderate-Autorisierung für US-Regionen. AWS East/West Einzelheiten zu den Compliance-Maßnahmen AWS und den Bemühungen zur Einhaltung der Vorschriften finden Sie unter AWS Services im Umfang nach Compliance-Programmen.

Wenn Sie mit AWS Services noch nicht vertraut sind, finden Sie in den folgenden Ressourcen weitere Informationen:

- AWS bietet Dienste für Datenverarbeitung, Datenbanken, Speicherung, Analyse und andere Funktionen. Eine Übersicht über alle AWS Services finden Sie unter Cloud Computing with Amazon Web Services.
- AWS bietet eine Reihe von Datenbankdiensten. Hinweise dazu, welcher Dienst für Ihre Umgebung am besten geeignet ist, finden Sie unter Datenbanken auf AWS.

Cluster

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



Instanzbasierte Amazon DocumentDB 5.0-Cluster unterstützen zwei Speicherkonfigurationen für einen Datenbankcluster: Amazon DocumentDB Standard und Amazon DocumentDB I/O-optimiert. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Speicherkonfigurationen.

Instances

Eine Amazon DocumentDB DocumentDB-Instance ist eine isolierte Datenbankumgebung in der Cloud. Eine Instance kann mehrere von Benutzern erstellte Datenbanken enthalten. Sie können eine Instance mit dem AWS Management Console oder dem AWS CLI erstellen und ändern.

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

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

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

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

> **Note**
>
> Eine vollständige Liste der Instance-Typen, die von Amazon DocumentDB in jeder Instance-Klasse unterstützt werden, finden Sie unterSpezifikationen der Instanzklasse.

Unterstützte Instance-Klassen nach Region

Region	R6GD	R6G	R5	R4	T4G	T3
USA Ost (Ohio)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA Ost (Nord-Virginia)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA West (Oregon)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Afrika (Kapstadt)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Südamerika (São Paulo)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hongkong)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hyderabad)			Unterstützt			Unterstützt
Asien-Pazifik (Mumbai)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt

Region	R6GD	R6G	R5	R4	T4G	T3
Asien-Pazifik (Seoul)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Sydney)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Singapur)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Tokio)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Kanada (Zentral)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Frankfurt)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Irland)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Europa (London)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Milan)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Paris)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Spain)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Stockholm)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Naher Osten (VAE)		Unterstützt	Unterstützt		Unterstützt	Unterstützt

Region	R6GD	R6G	R5	R4	T4G	T3
China (Peking)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
China (Ningxia)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Israel (Tel Aviv)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (US-West)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (US-Ost)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt

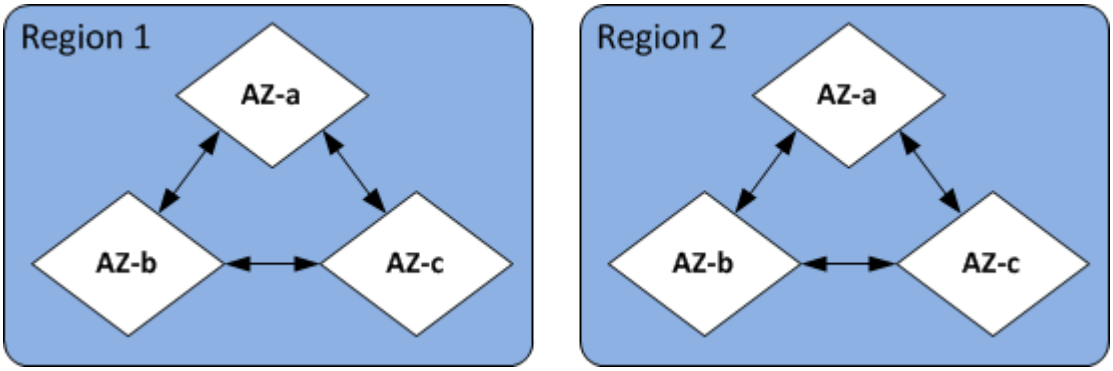
Regionen und Availability Zones

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

Regionen

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

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



Verfügbarkeitszonen

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

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

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

Name der Region	Region	Availability Zones (Compute)
USA Ost (Ohio)	`us-east-2`	3
USA Ost (Nord-Virginia)	`us-east-1`	6
USA West (Oregon)	`us-west-2`	4
Afrika (Kapstadt)	`af-south-1`	3

Name der Region	Region	Availability Zones (Compute)
Südamerika (São Paulo)	sa-east-1	3
Asien-Pazifik (Hongkong)	ap-east-1	3
Asien-Pazifik (Hyderabad)	ap-south-2	3
Asien-Pazifik (Mumbai)	ap-south-1	3
Asien-Pazifik (Seoul)	ap-northeast-2	4
Asien-Pazifik (Singapur)	ap-southeast-1	3
Asien-Pazifik (Sydney)	ap-southeast-2	3
Asien-Pazifik (Tokio)	ap-northeast-1	3
Kanada (Zentral)	ca-central-1	3
Region China (Peking)	cn-north-1	3
China (Ningxia)	cn-northwest-1	3
Europa (Frankfurt)	eu-central-1	3
Europa (Irland)	eu-west-1	3
Europa (London)	eu-west-2	3
Europa (Milan)	eu-south-1	3
Europa (Paris)	eu-west-3	3

Name der Region	Region	Availability Zones (Compute)
Europa (Spain)	`eu-south-2`	3
Europa (Stockholm)	`eu-north-1`	3
Naher Osten (VAE)	`me-central-1`	3
Israel (Tel Aviv)	`il-central-1`	3
AWS GovCloud (USA West)	`us-gov-west-1`	3
AWS GovCloud (US-Ost)	`us-gov-east-1`	3

Amazon DocumentDB — Preise

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

- Instance-Stunden (pro Stunde) — Basierend auf der Instance-Klasse der Instance (z. B.). `db.r5.xlarge` Die Preise werden auf Stundenbasis aufgeführt, aber Rechnungen werden jetzt auf die Sekunde genau kalkuliert und zeigen die Zeiten im Dezimalformat an. Die Nutzung von Amazon DocumentDB wird in Sekundenschritten mit einer Mindestdauer von 10 Minuten abgerechnet. Weitere Informationen finden Sie unter Instanzklassen verwalten.
- I/O-Anfragen (pro 1 Million Anfragen pro Monat) — Gesamtzahl der I/O Speicheranforderungen, die Sie in einem Abrechnungszeitraum stellen.
- Backup-Speicher (pro GiB pro Monat) — Backup-Speicher ist der Speicher, der automatisierten Datenbank-Backups und allen aktiven Datenbank-Snapshots, die Sie erstellt haben, zugeordnet ist. Wenn Sie die Aufbewahrungszeit Ihrer Backups erhöhen oder zusätzliche Datenbank-Snapshots erstellen, belegt Ihre Datenbank dementsprechend mehr Backup-Speicher. Der Backup-Speicher wird in GB-Monaten abgerechnet, die sekundengenaue Abrechnung wird hier nicht angewandt. Weitere Informationen finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.
- Datenübertragung (pro GB) — Datenübertragung innerhalb und aus Ihrer Instance vom oder ins Internet oder in andere AWS Regionen.

Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Preise.

Kostenlose Testversion

Sie können Amazon DocumentDB mit der einmonatigen kostenlosen Testversion kostenlos testen. Weitere Informationen finden Sie unter Kostenlose Testversion in den Preisen von Amazon DocumentDB oder in den häufig gestellten Fragen zur kostenlosen Amazon DocumentDB-Testversion.

Überwachen

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

Weitere Informationen finden Sie hier:

- Überwachen von Amazon DocumentDB mit CloudWatch
- Protokollieren Amazon DocumentDB DocumentDB-API-Aufrufen mit AWS CloudTrail

Schnittstellen

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

AWS Management Console

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

AWS CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Ihre Amazon DocumentDB-Cluster und -Instances zu verwalten. Mit minimaler Konfiguration können Sie

beginnen, alle Funktionen der Amazon DocumentDB DocumentDB-Konsole von Ihrem bevorzugten Terminalprogramm aus zu nutzen.

- Informationen zur AWS CLI Installation von finden Sie unter Installation der AWS Befehlszeilenschnittstelle.
- Informationen zur Verwendung von AWS CLI für Amazon DocumentDB finden Sie unter AWS Command Line Interface Reference for Amazon DocumentDB.

MongoDB-Treiber

Für die Entwicklung und das Schreiben von Anwendungen für einen Amazon DocumentDB-Cluster können Sie die MongoDB-Treiber auch mit Amazon DocumentDB verwenden. Weitere Informationen finden Sie auf der Registerkarte MongoDB-Shell unter Verbindung bei aktiviertem TLS herstellen oderVerbindung bei deaktiviertem TLS herstellen.

Als nächstes

In den vorangegangenen Abschnitten wurden Sie mit den grundlegenden Infrastrukturkomponenten von Amazon DocumentDB vertraut gemacht. Was sollten Sie als nächstes tun? Je nach Ihren Umständen finden Sie einen Einstieg in eines der folgenden Themen:

- Beginnen Sie mit Amazon DocumentDB, indem Sie einen Cluster und eine Instance mithilfe AWS CloudFormation Amazon DocumentDB DocumentDB-Schnellstart mit AWS CloudFormation von erstellen.
- Beginnen Sie mit Amazon DocumentDB, indem Sie einen Cluster und eine Instance anhand der Anweisungen in unserem Leitfaden für die ersten Schritte erstellen.
- Beginnen Sie mit Amazon DocumentDB, indem Sie mithilfe der Anweisungen unter einen elastischen Cluster erstellen. Erste Schritte mit Amazon DocumentDB Elastic Clusters
- Migrieren Sie Ihre MongoDB-Implementierung zu Amazon DocumentDB, indem Sie die Anleitung unter Migration zu Amazon DocumentDB

Amazon DocumentDB: So funktioniert's

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein vollständig verwalteter, MongoDB-kompatibler Datenbankservice. Mit Amazon DocumentDB können Sie denselben Anwendungscode

ausführen und dieselben Treiber und Tools verwenden, die Sie mit MongoDB verwenden. Amazon DocumentDB ist mit MongoDB 3.6, 4.0 und 5.0 kompatibel.

Themen

- Amazon DocumentDB DocumentDB-Endpunkte
- TLS-Unterstützung
- Amazon DocumentDB DocumentDB-Speicher
- Amazon DocumentDB DocumentDB-Replikation
- Zuverlässigkeit von Amazon DocumentDB
- Lesen Sie die Einstellungsoptionen
- TTL löscht
- Abrechnungsfähige Ressourcen

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

Ein Amazon DocumentDB-Cluster besteht aus zwei Komponenten:

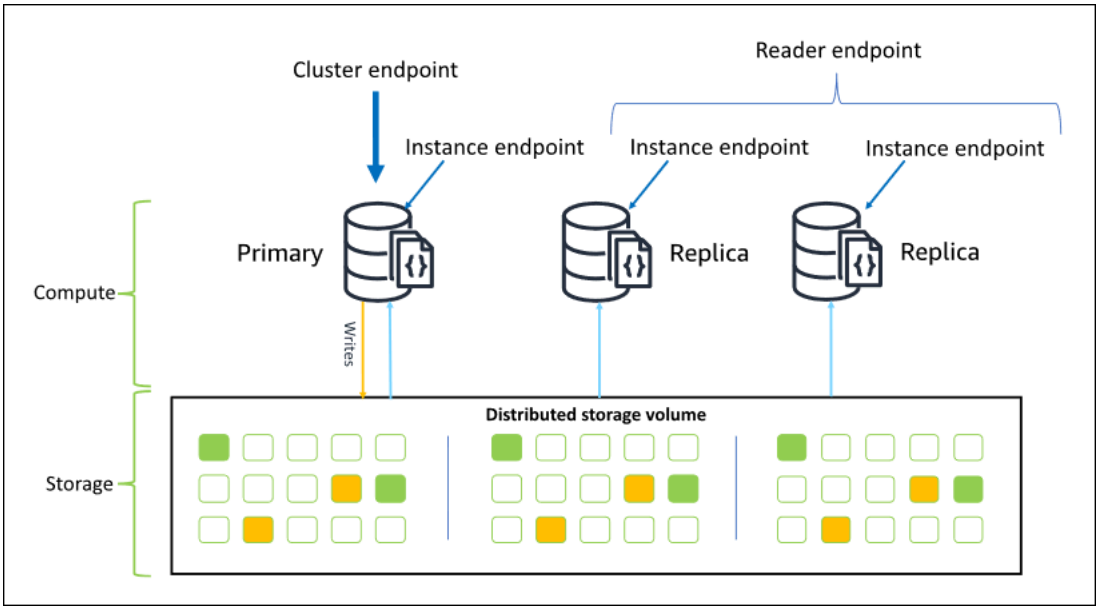
- Cluster-Volume — Verwendet einen Cloud-nativen Speicherservice, um Daten auf sechs Arten über drei Availability Zones hinweg zu replizieren und bietet so äußerst beständigen und verfügbaren Speicher. Ein Amazon DocumentDB-Cluster hat genau ein Cluster-Volume, das bis zu 128 TiB an Daten speichern kann.
- Instances — Stellen die Rechenleistung für die Datenbank bereit, indem sie Daten auf das Cluster-Speichervolume schreiben und Daten aus dem Cluster-Speichervolume lesen. Ein Amazon DocumentDB-Cluster kann 0—16 Instances haben.

Instances erfüllen eine von zwei Rollen:

- Primäre Instance — Unterstützt Lese- und Schreibvorgänge und führt alle Datenänderungen am Cluster-Volume durch. Jeder Amazon DocumentDB-Cluster hat eine primäre Instance.
- Replica-Instance — Unterstützt nur Lesevorgänge. Ein Amazon DocumentDB-Cluster kann zusätzlich zur primären Instance bis zu 15 Replikate haben. Die Verwendung mehrerer Replikate

ermöglicht es Ihnen, die Leseauslastungen zu verteilen. Darüber hinaus erhöhen Sie durch die Platzierung von Replikaten in separaten Availability Zones auch die Cluster-Verfügbarkeit.

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



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

Wenn Ihre Anwendung Daten in die Primär-Instance schreibt, schreibt diese die Daten dauerhaft in das Cluster-Volume. Anschließend repliziert es den Status dieses Schreibvorgangs (nicht die Daten) auf jedes aktive Replikat. Amazon DocumentDB DocumentDB-Replikate sind nicht an der Verarbeitung von Schreibvorgängen beteiligt, weshalb Amazon DocumentDB DocumentDB-Replikate für die Leseskalierung von Vorteil sind. Lesevorgänge von Amazon DocumentDB DocumentDB-Repliken sind letztlich konsistent mit minimaler Replikatverzögerung — in der Regel weniger als 100 Millisekunden, nachdem die primäre Instance die Daten geschrieben hat. Lesezugriffe von den Replikaten werden garantiert in der Reihenfolge gelesen, in der sie auf die primäre Instance geschrieben wurden. Die Replikationsverzögerung hängt von der Rate der Datenänderung ab. Perioden mit hoher Schreibaktivität können die Replikationsverzögerung erhöhen. Weitere Informationen finden Sie in den `ReplicationLag`-Metriken unter Amazon DocumentDB-Metriken.

Amazon DocumentDB DocumentDB-Endpunkte

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

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

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

Cluster-Endpunkt

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

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

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

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

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

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

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

Leser-Endpunkt

Der Reader-Endpunkt agiert als Load-Balancer für schreibgeschützte Verbindungen für alle verfügbaren Replikate in Ihrem Cluster. Ein Cluster-Reader-Endpunkt fungiert als Cluster-Endpunkt, wenn Sie eine Verbindung über den `replicaSet` Modus herstellen, d. h. in der Verbindungszeichenfolge lautet der Replikatsatzparameter. `&replicaSet=rs0` In diesem Fall können Sie Schreibvorgänge auf der Primärseite ausführen. Wenn Sie jedoch eine Verbindung zu dem angegebenen Cluster herstellen`directConnection=true`, führt der Versuch, einen Schreibvorgang über eine Verbindung zum Leser-Endpunkt auszuführen, zu einem Fehler. Ein Amazon DocumentDB-Cluster hat genau einen Leser-Endpunkt.

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

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

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

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

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

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

Clusterendpunkt als Replikatsatz herstellen und die Lesevorstellungsoption secondaryPreferred nutzen.

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

Instance-Endpunkt

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

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

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

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

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

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

> **Note**
>
> Die Rolle einer Instance als "Primär" oder "Replikat" kann sich aufgrund eines Failover-Ereignisses ändern. Ihre Anwendungen sollten niemals davon ausgehen, dass ein bestimmter Instance-Endpunkt die primäre Instance ist. Es wird nicht empfohlen, eine Verbindung zu Instance-Endpunkten für Produktionsanwendungen herzustellen. Stattdessen

wird empfohlen, dass Sie mithilfe des Clusterendpunkts und im Replikatsatzmodus eine Verbindung zum Cluster herstellen (siehe Als Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen). Weitere Informationen zur erweiterten Kontrolle der Instance-Failover-Priorität finden Sie unter Grundlegendes zur Amazon DocumentDB-Cluster-Fehlertoleranz.

Informationen zum Suchen der Endpunkte eines Clusters finden Sie unter Den Endpunkt einer Instanz finden.

Replikat-Set-Modus

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

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

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

Wenn Sie eine Verbindung im Replikatgruppenmodus herstellen, wird Ihr Amazon DocumentDB-Cluster Ihren Treibern und Clients als Replikatsatz angezeigt. Instances, die Ihrem Amazon DocumentDB-Cluster hinzugefügt und daraus entfernt wurden, werden automatisch in der Konfiguration des Replikatsatzes wiedergegeben.

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

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

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

TLS-Unterstützung

Weitere Informationen zur Verbindung mit Amazon DocumentDB mithilfe von Transport Layer Security (TLS) finden Sie unterVerschlüsseln von Daten während der Übertragung.

Amazon DocumentDB DocumentDB-Speicher

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

Informationen zur Abrechnung des -Datenspeichers

Amazon DocumentDB erhöht automatisch die Größe eines Cluster-Volumes, wenn die Datenmenge zunimmt. Ein Amazon DocumentDB-Cluster-Volume kann auf eine maximale Größe von 128 TiB anwachsen. Ihnen wird jedoch nur der Speicherplatz in Rechnung gestellt, den Sie in einem Amazon DocumentDB-Cluster-Volume verwenden. Ab Amazon DocumentDB 4.0 verringert sich der zugewiesene Speicherplatz um einen vergleichbaren Betrag, wenn Daten entfernt werden, z. B. durch Löschen einer Sammlung oder eines Indexes. Somit können Sie die Speichergebühren senken, indem Sie Sammlungen, Indizes und Datenbanken löschen, die Sie nicht mehr benötigen. In Amazon DocumentDB Version 3.6 kann das Cluster-Volume Speicherplatz wiederverwenden, der beim Entfernen von Daten frei wird, aber das Volume selbst nimmt nie an Größe ab. Daher werden Sie in Version 3.6 möglicherweise keine Änderung des Speicherplatzes feststellen, wenn Sie eine Sammlung oder einen Index löschen, obwohl der freigewordene Speicherplatz wiederverwendet wird.

> **Note**
>
> Bei Amazon DocumentDB 3.6 basieren die Speicherkosten auf der Speichergrenze (der Höchstmenge, die dem Amazon DocumentDB-Cluster zu einem beliebigen Zeitpunkt zugewiesen wurde). Sie können die Kosten kontrollieren, indem Sie ETL-Praktiken vermeiden, die große Mengen temporärer Informationen erzeugen oder große Mengen neuer Daten laden, bevor nicht benötigte ältere Daten entfernt werden. Wenn das Entfernen von Daten aus einem Amazon DocumentDB-Cluster dazu führt, dass eine beträchtliche

Menge an zugewiesenem, aber ungenutztem Speicherplatz zur Verfügung steht, muss zum Zurücksetzen der Höchstgrenze ein logischer Datendump und eine Wiederherstellung auf einem neuen Cluster mit einem Tool wie oder durchgeführt werden. `mongodump` `mongorestore` Das Erstellen und Wiederherstellen eines Snapshots führt nicht zur Reduzierung des zugeteilten Speichers, da das physische Layout des zugrunde liegenden Speichers im wiederhergestellten Snapshot unverändert bleibt.

Note

Für die Nutzung von `mongodump` Hilfsprogrammen `mongorestore` fallen I/O Gebühren an, die von der Größe der Daten abhängen, die auf das Speichervolume gelesen und geschrieben werden.

Informationen zur Amazon DocumentDB-Datenspeicherung und zu den I/O Preisen finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) — Preise und Preise. FAQs

Amazon DocumentDB DocumentDB-Replikation

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

Zuverlässigkeit von Amazon DocumentDB

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

Automatische Speicherplatzreparatur

Amazon DocumentDB verwaltet mehrere Kopien Ihrer Daten in drei Availability Zones, wodurch das Risiko eines Datenverlusts aufgrund eines Speicherausfalls erheblich reduziert wird. Amazon DocumentDB erkennt automatisch Fehler im Cluster-Volume. Wenn ein Segment eines Cluster-Volumes ausfällt, repariert Amazon DocumentDB das Segment sofort. Es verwendet die Daten der anderen Volumes, aus denen sich das Cluster-Volumen zusammensetzt, um sicherzustellen, dass die Daten im reparierten Segment aktuell sind. Dadurch vermeidet Amazon DocumentDB Datenverluste und reduziert die Notwendigkeit, nach einem Instance-Ausfall eine point-in-time Wiederherstellung durchzuführen.

Überlebensfähiges Cache-Warming

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

Wiederherstellung nach einem Ausfall

Amazon DocumentDB ist so konzipiert, dass es nach einem Absturz fast sofort wiederhergestellt wird und Ihre Anwendungsdaten weiterhin bereitgestellt werden. Amazon DocumentDB führt die Wiederherstellung nach einem Absturz asynchron auf parallel Threads durch, sodass Ihre Datenbank nach einem Absturz fast unmittelbar geöffnet und verfügbar ist.

Verwaltung von Ressourcen

Amazon DocumentDB schützt Ressourcen, die für die Ausführung kritischer Prozesse im Service benötigt werden, wie z. B. Zustandsprüfungen. Zu diesem Zweck drosselt Amazon DocumentDB Anfragen, wenn eine Instance unter hohem Speicherdruck steht. Daher können einige Operationen in die Warteschlange gestellt werden, um darauf zu warten, dass der Speicherdruck nachlässt. Wenn die Speicherauslastung anhält, kann es bei Vorgängen in der Warteschlange zu einer Zeitüberschreitung kommen. Anhand der folgenden CloudWatch Messwerte können Sie überwachen, ob der Dienst aufgrund von zu wenig Arbeitsspeicher Drosselungen durchführt oder nicht:`LowMemThrottleQueueDepth`,. `LowMemThrottleMaxQueueDepth` `LowMemNumOperationsThrottled` `LowMemNumOperationsTimedOut` Weitere Informationen finden Sie unter Amazon DocumentDB überwachen mit CloudWatch. Wenn Sie aufgrund der

LowMem CloudWatch Metriken einen anhaltenden Speicherdruck auf Ihrer Instance feststellen, empfehlen wir Ihnen, Ihre Instance hochzuskalieren, um zusätzlichen Speicher für Ihre Arbeitslast bereitzustellen.

Lesen Sie die Einstellungsoptionen

Amazon DocumentDB verwendet einen Cloud-nativen Shared Storage-Service, der Daten sechsmal über drei Availability Zones hinweg repliziert, um ein hohes Maß an Haltbarkeit zu gewährleisten. Amazon DocumentDB ist nicht darauf angewiesen, Daten auf mehrere Instanzen zu replizieren, um Haltbarkeit zu erreichen. Die Daten Ihres Clusters sind beständig, unabhängig davon, ob sie eine einzelne Instance oder 15 Instances enthalten.

Themen

- Haltbarkeit beim Schreiben
- Isolierung lesen
- Lesekonsistenz
- Hohe Verfügbarkeit
- Lesevorgänge skalieren

Haltbarkeit beim Schreiben

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

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

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

Schreibvorgänge in einen Amazon DocumentDB-Cluster erfolgen innerhalb eines einzigen Dokuments atomar.

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

Isolierung lesen

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

Lesekonsistenz

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

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

Amazon DocumentDB Leseeinstellungen

Amazon DocumentDB unterstützt das Festlegen einer Lesepräferenzoption nur beim Lesen von Daten vom Cluster-Endpunkt im Replikatsatzmodus. Das Festlegen einer Lesepräferenzoption wirkt sich darauf aus, wie Ihr MongoDB-Client oder -Treiber Leseanfragen an Instances in Ihrem Amazon

DocumentDB-Cluster weiterleitet. Sie können Leseeinstellungen für eine bestimmte Abfrage oder als allgemeine Option in Ihrem MongoDB-Treiber festlegen. (Lesen Sie in der Dokumentation Ihres Clients oder Treibers nach, wie Sie eine Leseeinstellung festlegen können.)

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

Amazon DocumentDB unterstützt die Einstellung von Tag-Sets als Lesepräferenz nicht.

Unterstützte Leseeinstellungsoptionen

- **`primary`**— Durch die Angabe einer `primary` Lesepräferenz wird sichergestellt, dass alle Lesevorgänge an die primäre Instance des Clusters weitergeleitet werden. Wenn die primäre Instance nicht verfügbar ist, schlägt der Lesevorgang fehl. Eine `primary` Lesepräferenz sorgt für read-after-write Konsistenz und eignet sich für Anwendungsfälle, in denen read-after-write Konsistenz Vorrang vor Hochverfügbarkeit und Leseskalierung hat.

  Im folgenden Beispiel wird die Leseeinstellung `primary` angegeben:

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

- **`primaryPreferred`**— Die Angabe einer `primaryPreferred` Lesepräferenz leitet Lesevorgänge bei normalem Betrieb an die primäre Instance weiter. Wenn es ein primäres Failover gibt, leitet der Client Anfragen an ein Replikat weiter. Eine `primaryPreferred` Lesepräferenz sorgt für read-after-write Konsistenz während des normalen Betriebs und letztendlich für konsistente Lesevorgänge während eines Failover-Ereignisses. Eine `primaryPreferred` Lesepräferenz eignet sich für Anwendungsfälle, in denen read-after-write Konsistenz Vorrang vor Leseskalierung hat, aber dennoch eine hohe Verfügbarkeit erforderlich ist.

  Im folgenden Beispiel wird die Leseeinstellung `primaryPreferred` angegeben:

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

- **`secondary`**— Durch die Angabe einer `secondary` Lesepräferenz wird sichergestellt, dass Lesevorgänge nur an ein Replikat und niemals an die primäre Instanz weitergeleitet werden.

Wenn es in einem Cluster keine Replikat-Instances gibt, schlägt die Leseanforderung fehl. Eine `secondary` Lesepräferenz führt letztendlich zu konsistenten Lesevorgängen und eignet sich für Anwendungsfälle, in denen der Schreibdurchsatz der primären Instanz Vorrang vor hoher Verfügbarkeit und Konsistenz hat. read-after-write

Im folgenden Beispiel wird die Leseeinstellung `secondary` angegeben:

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

- **`secondaryPreferred`**— Durch die Angabe einer `secondaryPreferred` Lesepräferenz wird sichergestellt, dass Lesevorgänge an eine Read Replica weitergeleitet werden, wenn ein oder mehrere Replikate aktiv sind. Wenn es in einem Cluster keine aktiven Replikat-Instances gibt, wird die Leseanforderung an die primäre Instance weitergeleitet. Die Leseeinstellung `secondaryPreferred` ergibt Eventually Consistent-Lesezugriffe, wenn der Lesezugriff durch ein Read Replica bedient wird. Dadurch wird read-after-write Konsistenz gewährleistet, wenn der Lesevorgang von der primären Instanz verarbeitet wird (mit Ausnahme von Failover-Ereignissen). Eine `secondaryPreferred` Lesepräferenz eignet sich für Anwendungsfälle, in denen Leseskalierung und Hochverfügbarkeit Vorrang vor Konsistenz haben. read-after-write

  Im folgenden Beispiel wird die Leseeinstellung `secondaryPreferred` angegeben:

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

- **`nearest`**— Die Angabe einer `nearest` Lesepräferenz leitet Lesevorgänge ausschließlich auf der Grundlage der gemessenen Latenz zwischen dem Client und allen Instances im Amazon DocumentDB-Cluster weiter. Die Leseeinstellung `nearest` ergibt Eventually Consistent-Lesezugriffe, wenn der Lesezugriff durch ein Read Replica bedient wird. Dadurch wird read-after-write Konsistenz gewährleistet, wenn der Lesevorgang von der primären Instance bedient wird (mit Ausnahme von Failover-Ereignissen). Eine `nearest` Lesepräferenz eignet sich für Anwendungsfälle, in denen das Erreichen einer möglichst niedrigen Leselatenz und hoher Verfügbarkeit Vorrang vor read-after-write Konsistenz und Leseskalierung hat.

  Im folgenden Beispiel wird die Leseeinstellung `nearest` angegeben:

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

Hohe Verfügbarkeit

Amazon DocumentDB unterstützt hochverfügbare Cluster-Konfigurationen, indem Repliken als Failover-Ziele für die primäre Instance verwendet werden. Wenn die primäre Instance ausfällt, wird ein Amazon DocumentDB DocumentDB-Replikat zur neuen primären Instance hochgestuft, mit einer kurzen Unterbrechung, während der Lese- und Schreibanforderungen an die primäre Instance mit einer Ausnahme fehlschlagen.

Wenn Ihr Amazon DocumentDB-Cluster keine Replikate enthält, wird die primäre Instance bei einem Ausfall neu erstellt. Das Heraufstufen eines Amazon DocumentDB DocumentDB-Replikats ist jedoch viel schneller als das Neuerstellen der primären Instance. Wir empfehlen daher, ein oder mehrere Amazon DocumentDB DocumentDB-Replikate als Failover-Ziele zu erstellen.

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

Lesevorgänge skalieren

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

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

Amazon DocumentDB unterstützt die clientseitige Leseskalierung mithilfe von Read Preference-Optionen. Weitere Informationen finden Sie unter Amazon DocumentDB Leseeinstellungen.

TTL löscht

Löschungen aus einem TTL-Indexbereich über einen Hintergrundprozess erfolgen nach dem Best-Effort-Prinzip und können nicht für einen bestimmten Zeitrahmen garantiert werden. Faktoren wie

Instance-Größe, Instance-Ressourcenauslastung, Dokumentgröße und Gesamtdurchsatz können sich auf den Zeitpunkt einer TTL-Löschung auswirken.

Wenn der TTL-Monitor Ihre Dokumente löscht, entstehen bei jeder Löschung E/A-Kosten, was den Rechnungsbetrag erhöht. Wenn die Durchsatz- und TTL-Löschraten steigen, sollten Sie aufgrund der erhöhten I/O-Auslastung mit einer Erhöhung Ihrer Rechnung rechnen.

Wenn Sie einen TTL-Index für eine bestehende Sammlung erstellen, müssen Sie alle abgelaufenen Dokumente löschen, bevor Sie den Index erstellen. Die aktuelle TTL-Implementierung ist für das Löschen eines kleinen Teils der Dokumente in der Sammlung optimiert. Dies ist typisch, wenn TTL für die Sammlung von Anfang an aktiviert war. Dies kann zu höheren IOPS als nötig führen, wenn eine große Anzahl von Dokumenten auf einmal gelöscht werden muss.

Wenn Sie keinen TTL-Index zum Löschen von Dokumenten erstellen möchten, können Sie Dokumente stattdessen nach Zeit in Sammlungen unterteilen und diese Sammlungen einfach löschen, wenn die Dokumente nicht mehr benötigt werden. Beispiel: Sie können eine Sammlung pro Woche erstellen und diese löschen, ohne dass IO-Kosten anfallen. Dies kann deutlich kostengünstiger sein als die Verwendung eines TTL-Index.

Abrechnungsfähige Ressourcen

Identifizieren von kostenpflichtigen Amazon DocumentDB DocumentDB-Ressourcen

Als vollständig verwalteter Datenbankservice berechnet Amazon DocumentDB Gebühren für Instances, Speicher, I/Os, Backups und Datenübertragung. Weitere Informationen finden Sie unter Preise für Amazon DocumentDB (mit MongoDB-Kompatibilität).

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

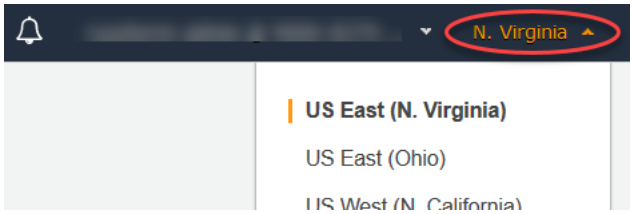
Verwenden Sie den AWS Management Console

Mithilfe von können Sie die AWS Management Console Amazon DocumentDB-Cluster, -Instances und -Snapshots ermitteln, die Sie für eine bestimmte Person bereitgestellt haben. AWS-Region

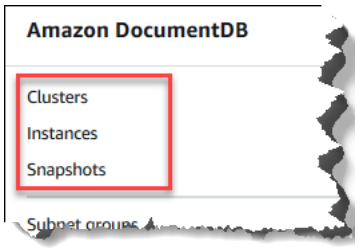
So ermitteln Sie Cluster, Instances und Snapshots:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Um nach abrechnungsfähigen Ressourcen in einer anderen Region als Ihrer Standardregion zu suchen, wählen Sie in der oberen rechten Ecke des Bildschirms die Region aus, nach der Sie suchen möchten. AWS-Region



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



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

Mit dem AWS CLI

Mithilfe von können Sie die AWS CLI Amazon DocumentDB-Cluster, -Instances und -Snapshots ermitteln, die Sie für eine bestimmte Person bereitgestellt haben. AWS-Region

So ermitteln Sie Cluster und Instances:

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

Example

Für Linux, macOS oder Unix:

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

Für Windows:

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

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

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

So ermitteln Sie Snapshots:

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

Für Linux, macOS oder Unix:

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

Für Windows:

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

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

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

```
    [
        "test-snap",
        "manual"
    ]
]
```

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

Löschen unerwünschter kostenpflichtiger Ressourcen

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

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

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

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

Was ist eine Dokumentendatenbank?

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

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

Themen

- Anwendungsfälle für Dokumentendatenbanken
- Dokumente verstehen
- Arbeiten mit Dokumenten

Anwendungsfälle für Dokumentendatenbanken

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

Themen

- Benutzerprofile
- Big Data in Echtzeit
- Verwaltung von Inhalten

Benutzerprofile

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

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

Big Data in Echtzeit

In der Vergangenheit wurde die Fähigkeit, Informationen aus Betriebsdaten zu extrahieren, dadurch beeinträchtigt, dass operative Datenbanken und analytische Datenbanken in unterschiedlichen Umgebungen verwaltet wurden, d. h. in Bezug auf Betrieb und Geschäfts-/Berichtswesen. Die Fähigkeit, operative Informationen in Echtzeit zu extrahieren, ist in einem

hart umkämpften Geschäftsumfeld von entscheidender Bedeutung. Durch die Verwendung von Dokumentendatenbanken kann ein Unternehmen Betriebsdaten aus jeder beliebigen Quelle speichern und verwalten und gleichzeitig die Daten zur Analyse an die BI-Engine seiner Wahl weiterleiten. Es ist nicht erforderlich, dass zwei Umgebungen vorhanden sein müssen.

Verwaltung von Inhalten

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

Dokumente verstehen

Dokumentendatenbanken werden verwendet, um semistrukturierte Daten als Dokument zu speichern, anstatt Daten in mehreren Tabellen zu normalisieren, von denen jede eine eindeutige und feste Struktur hat, wie in einer relationalen Datenbank. Dokumente, die in einer Dokumentdatenbank gespeichert sind, verwenden verschachtelte Schlüssel-Werte-Paare, um die Struktur oder das Schema des Dokuments bereitzustellen. Es können jedoch verschiedene Arten von Dokumenten in derselben Dokumentdatenbank gespeichert werden, wodurch die Anforderung erfüllt wird, ähnliche Daten in unterschiedlichen Formaten zu verarbeiten. Da beispielsweise jedes Dokument selbstbeschreibend ist, können die in diesem Abschnitt beschriebenen JSON-kodierten Dokumente für einen Online-Shop, die unter dem Thema Beispieldokumente in einer Dokumentendatenbank beschrieben werden, in derselben Dokumentendatenbank gespeichert werden.

Themen

- SQL im Vergleich zu nichtrelationaler Terminologie
- Einfache Dokumente
- Eingebettete Dokumente
- Beispieldokumente in einer Dokumentendatenbank
- Grundlegendes zur Normalisierung in einer Dokumentendatenbank

SQL im Vergleich zu nichtrelationaler Terminologie

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

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

Einfache Dokumente

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

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

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

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

Eingebettete Dokumente

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

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

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

Beispieldokumente in einer Dokumentendatenbank

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

Im Folgenden finden Sie ein Beispiel-Buch-Dokument:

```
{
```

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

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

```
{
    "_id" : "0123456789012",
    "Publication": "Programming Today",
    "Issue":
    {
        "Volume": "14",
        "Number": "09"
    },
    "Articles" : [
        {
            "Title": "Is a Document Database Your Best Solution?",
            "Author":
            {
                "LName": "Major",
                "FName": "Mary"
            }
        },
        {
            "Title": "Databases for Online Solutions",
            "Author":
            {
                "LName": "Stiles",
                "FName": "John"
            }
        }
    ],
    "Type": "periodical"
}
```

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

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

Grundlegendes zur Normalisierung in einer Dokumentendatenbank

Dokumentdatenbanken werden nicht normalisiert; in einem Dokument gefundene Daten können in einem anderen Dokument wiederholt werden. Darüber hinaus können einige Datenunterschiede zwischen den Dokumenten bestehen. Betrachten Sie beispielsweise das Szenario, in dem Sie einen Einkauf in einem Online-Shop tätigen und alle Details Ihrer Einkäufe in einem einzigen Dokument gespeichert sind. Das Dokument könnte etwa so aussehen wie das folgende JSON-Dokument:

```
{
    "DateTime": "2018-08-15T12:13:10Z",
    "LName" : "Santos",
    "FName" : "Paul",
    "Cart" : [
        {
            "ItemId" : "9876543210123",
            "Description" : "Understanding Document Databases",
            "Price" : "29.95"
        },
        {
            "ItemId" : "0123456789012",
            "Description" : "Programming Today",
            "Issue": {
                "Volume": "14",
                "Number": "09"
            },
            "Price" : "8.95"
        },
        {
            "ItemId": "234567890-K",
```

```
            "Description": "Gel Pen (black)",
            "Price": "2.49"
        }
    ],
    "PaymentMethod" :
    {
        "Issuer" : "MasterCard",
        "Number" : "1234-5678-9012-3456"
    },
    "ShopperId" : "1234567890"
}
```

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

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

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

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

Arbeiten mit Dokumenten

Als Dokumentendatenbank macht es Amazon DocumentDB einfach, JSON-Daten zu speichern, abzufragen und zu indizieren. In Amazon DocumentDB entspricht eine Sammlung einer Tabelle in einer relationalen Datenbank, außer dass kein einziges Schema für alle Dokumente durchgesetzt wird. Mit Sammlungen können Sie ähnliche Dokumente gruppieren und gleichzeitig alle in derselben Datenbank halten, ohne dass sie in ihrer Struktur identisch sein müssen.

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

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

Themen

- Dokumente hinzufügen
- Dokumente werden abgefragt
- Dokumente werden aktualisiert
- Dokumente löschen

Dokumente hinzufügen

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

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

Themen

- Ein einzelnes Dokument hinzufügen
- Hinzufügen mehrerer Dokumente

Ein einzelnes Dokument hinzufügen

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

```
db.example.insertOne(
    {
        "Item": "Ruler",
        "Colors": ["Red","Green","Blue","Clear","Yellow"],
        "Inventory": {
            "OnHand": 47,
            "MinOnHand": 40
        },
        "UnitPrice": 0.89
    }
)
```

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

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

Hinzufügen mehrerer Dokumente

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

```
db.example.insertMany(
    [
```

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

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

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

Dokumente werden abgefragt

Manchmal müssen Sie möglicherweise den Bestand Ihres Online-Shops nachschlagen, damit Kunden das Angebot sehen und kaufen können. Die Abfrage einer Sammlung ist relativ einfach,

unabhängig davon, ob Sie alle Dokumente in der Sammlung haben möchten oder nur die Dokumente, die ein bestimmtes Kriterium erfüllen.

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

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

Themen

- Alle Dokumente in einer Sammlung abrufen
- Dokumente werden abgerufen, die einem Feldwert entsprechen
- Dokumente werden abgerufen, die einem eingebetteten Dokument entsprechen
- Dokumente werden abgerufen, die einem Feldwert in einem eingebetteten Dokument entsprechen
- Dokumente werden abgerufen, die einem Array entsprechen
- Dokumente werden abgerufen, die einem Wert in einem Array entsprechen
- Dokumente mithilfe von Operatoren abrufen

Alle Dokumente in einer Sammlung abrufen

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

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

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

Dokumente werden abgerufen, die einem Feldwert entsprechen

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

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

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

Dokumente werden abgerufen, die einem eingebetteten Dokument entsprechen

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

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

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

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

Dokumente werden abgerufen, die einem Feldwert in einem eingebetteten Dokument entsprechen

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

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

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

Dokumente werden abgerufen, die einem Array entsprechen

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

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

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

Dokumente werden abgerufen, die einem Wert in einem Array entsprechen

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

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

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

Dokumente mithilfe von Operatoren abrufen

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

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

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

Dokumente werden aktualisiert

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

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

Wenn Sie ein vorhandenes Feld aktualisieren — unabhängig davon, ob es sich bei dem Feld um ein einfaches Feld, ein Array oder ein eingebettetes Dokument handelt — geben Sie den Feldnamen und seine Werte an. Am Ende der Operation wirkt es so, als ob das Feld im alten Dokument durch das neue Feld und die Werte ersetzt wurde.

Themen

- Aktualisierung der Werte eines vorhandenen Felds
- Ein neues Feld hinzufügen
- Ein eingebettetes Dokument ersetzen
- Neue Felder in ein eingebettetes Dokument einfügen
- Ein Feld aus einem Dokument entfernen
- Ein Feld aus mehreren Dokumenten entfernen

Aktualisierung der Werte eines vorhandenen Felds

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

```
{
    "Item": "Ruler",
    "Colors": ["Red","Green","Blue","Clear","Yellow"],
    "Inventory": {
        "OnHand": 47,
        "MinOnHand": 40
    },
    "UnitPrice": 0.89
},
{
    "Item": "Pen",
    "Colors": ["Red","Green","Blue","Black"],
    "Inventory": {
        "OnHand": 244,
        "MinOnHand": 72
```

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

So aktualisieren Sie ein einfaches Feld

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

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

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

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

So aktualisieren Sie ein Array

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

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

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

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

Ein neues Feld hinzufügen

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

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

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

Die Ergebnisse dieser Operation sehen ungefähr wie folgt aus (JSON-Format).

```
{
    "Item": "Spray Paint",
    "Colors": ["Black","Red","Green","Blue"],
    "Inventory": {
        "OnHand": 47,
```

```
        "MinOnHand": 50,
        "OrderQnty": 36
    },
    "UnitPrice": 3.99
}
```

Ein eingebettetes Dokument ersetzen

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

Anhand des folgenden Dokuments

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

So ersetzen Sie ein eingebettetes Dokument

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

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

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

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

```
}
```

Neue Felder in ein eingebettetes Dokument einfügen

So fügen Sie einem eingebetteten Dokument neue Felder hinzu

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

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

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

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

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

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

Ein Feld aus einem Dokument entfernen

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

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

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

Ergebnisse dieser Operation sehen in etwa folgendermaßen aus.

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

Ein Feld aus mehreren Dokumenten entfernen

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

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

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

```
)
```

Dokumente löschen

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

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

Um alle Dokumente aus Ihrer Datenbank zu entfernen, verwenden Sie den `remove()` Vorgang mit einer leeren Abfrage.

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

Erste Schritte mit Amazon DocumentDB

Es gibt viele Möglichkeiten, eine Verbindung herzustellen und mit Amazon DocumentDB zu beginnen. Dieses Handbuch ist der schnellste, einfachste und einfachste Weg für Benutzer, mit unserer leistungsstarken Dokumentendatenbank zu beginnen. In diesem Handbuch wird verwendet AWS CloudShell, um eine Verbindung herzustellen und Ihren Amazon DocumentDB-Cluster direkt vom AWS Management Console abzufragen. Neukunden, die Anspruch auf das AWS kostenlose Kontingent haben, können Amazon DocumentDB CloudShell kostenlos nutzen. Wenn Ihre AWS CloudShell Umgebung oder Ihr Amazon DocumentDB-Cluster Ressourcen nutzt, die über das kostenlose Kontingent hinausgehen, werden Ihnen die normalen AWS Tarife für diese Ressourcen berechnet. Mit diesem Handbuch können Sie in weniger als fünf Minuten mit Amazon DocumentDB beginnen.

> **Note**
>
> Die Anweisungen in diesem Handbuch beziehen sich speziell auf das Erstellen und Herstellen einer Verbindung zu instanzbasierten Amazon DocumentDB-Clustern, in denen Amazon DocumentDB und verfügbar sind. AWS CloudShell
>
> - Informationen dazu, wie Sie elastische Amazon DocumentDB-Cluster erstellen und eine Verbindung zu diesen herstellen möchten, finden Sie unterErste Schritte mit Amazon DocumentDB Elastic Clusters.
> - Wenn Sie in Regionen AWS China ansässig sind, finden Sie weitere Informationen unterAmazon EC2 automatisch Connect.

Themen

- Voraussetzungen
- Schritt 1: Erstellen eines Clusters
- Schritt 2: Connect zu Ihrem Cluster her
- Schritt 3: Daten einfügen und abfragen
- Schritt 4: Erkunden

Voraussetzungen

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

Erstellen Sie ein Amazon Web Services (AWS) -Konto

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

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

Um sich für eine anzumelden AWS-Konto

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

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

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

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

Für den Zugriff auf die Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen wie Clustern, Instances und Cluster-Parametergruppen sind Anmeldeinformationen erforderlich, mit denen Sie Ihre Anfragen authentifizieren AWS können. Weitere Informationen finden Sie unter Identity and Access Management für Amazon DocumentDB.

1. Geben Sie in der Suchleiste von IAM ein und wählen Sie im angezeigten Drop-down-Menü IAM aus. AWS Management Console
2. Sobald Sie sich in der IAM-Konsole befinden, wählen Sie im Navigationsbereich Benutzer aus.
3. Wählen Sie Ihren Nutzernamen aus.

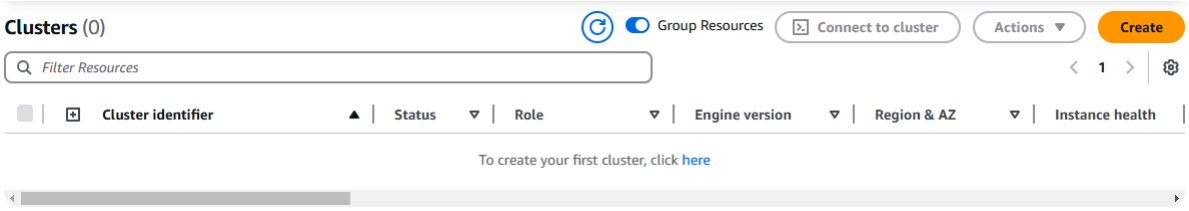
4. Klicken Sie auf Berichtigung hinzufügen.
5. Wählen Sie Richtlinien direkt anhängen aus.
6. Geben Sie `AmazonDocDBFullAccess` in die Suchleiste ein und wählen Sie sie aus, sobald sie in den Suchergebnissen erscheint.
7. Klicken Sie auf Weiter.
8. Klicken Sie auf Berichtigung hinzufügen.

> **Note**
>
> Ihr AWS Konto enthält in jeder Region eine Standard-VPC. Wenn Sie sich für die Verwendung einer Amazon VPC entscheiden, führen Sie die Schritte im Thema Amazon VPC erstellen im Amazon VPC-Benutzerhandbuch aus.

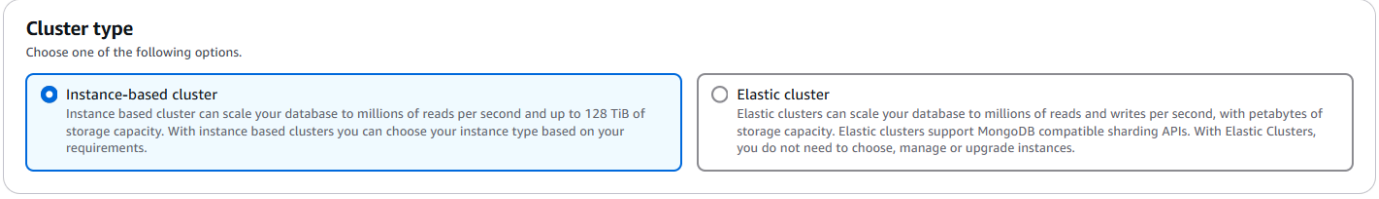
Schritt 1: Erstellen eines Clusters

In diesem Schritt erstellen Sie einen Amazon DocumentDB-Cluster.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Amazon DocumentDB-Managementkonsole unter Clusters die Option Create aus.



3. Wählen Sie auf der Seite Amazon DocumentDB-Cluster erstellen im Abschnitt Clustertyp die Option Instance-based cluster (dies ist die Standardoption).

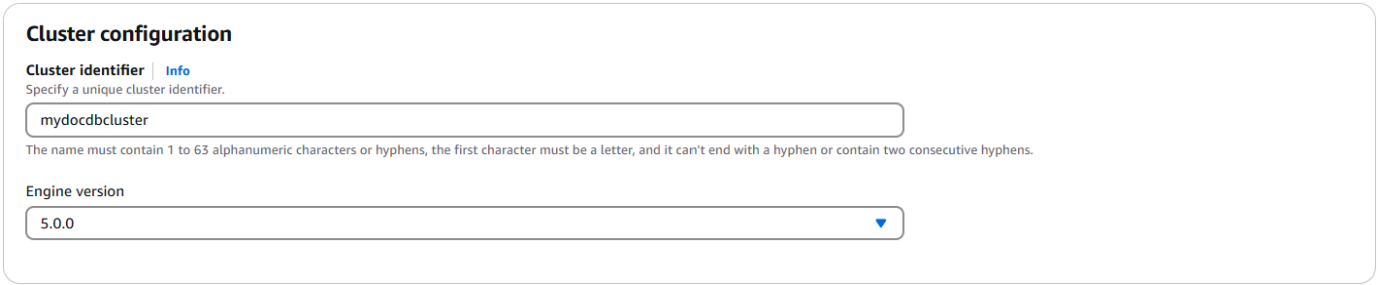


> **Note**
>
> Die andere Option in dieser Kategorie ist Elastic Cluster. Weitere Informationen zu Amazon DocumentDB Elastic Clusters finden Sie unter Verwendung elastischer Amazon DocumentDB-Cluster

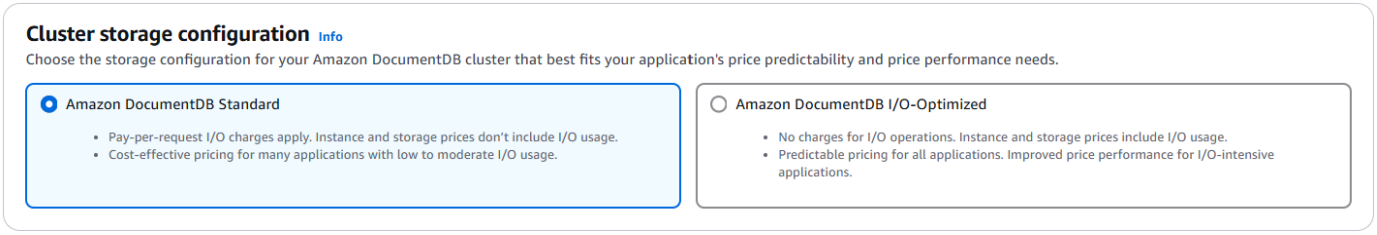
4. Im Abschnitt Cluster-Konfiguration:

   a. Geben Sie als Cluster-ID einen eindeutigen Namen ein, z. **`mydocdbcluster`** B. Beachten Sie, dass die Konsole alle Clusternamen unabhängig davon, wie sie eingegeben wurden, in Kleinbuchstaben ändert.

   b. Wählen Sie für Engine-Version 5.0.0 aus.



5. Wählen Sie im Abschnitt Cluster-Speicherkonfiguration Amazon DocumentDB Standard aus (dies ist die Standardoption).



> **Note**
>
> Die andere Option in dieser Kategorie ist Amazon DocumentDB I/O-Optimized. Weitere Informationen zu den beiden Optionen finden Sie unter Amazon DocumentDB-Cluster-Speicherkonfigurationen

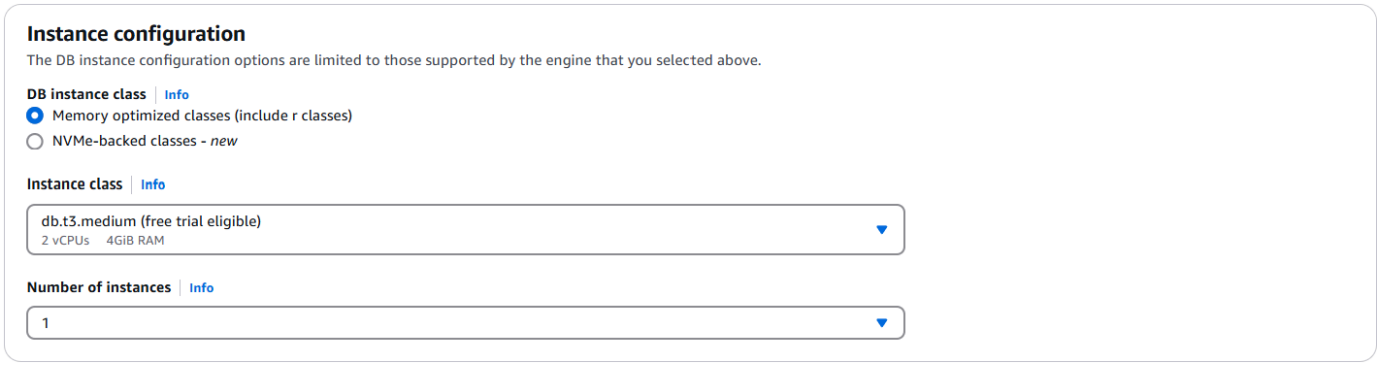
6. Im Abschnitt Instanzkonfiguration:

a. Wählen Sie für DB-Instance-Klasse die Option Speicheroptimierte Klassen (einschließlich R-Klassen) aus (dies ist die Standardeinstellung).

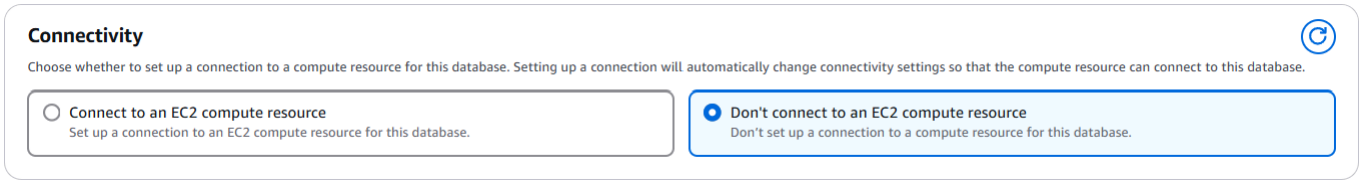
   Bei der anderen Instance-Option handelt es sich um Klassen mit NVMe -backed. Weitere Informationen hierzu finden Sie unter NVMe-gestützte Instances.

b. Wählen Sie für Instance-Klasse db.t3.medium aus. Dies ist für die kostenlose Testversion berechtigt. AWS

c. Wählen Sie für Anzahl der Instanzen die Option 1 Instanz aus. Die Auswahl einer Instanz trägt zur Kostenminimierung bei. Wenn es sich um ein Produktionssystem handeln würde, würden wir empfehlen, dass Sie drei Instanzen bereitstellen, um eine hohe Verfügbarkeit zu gewährleisten.

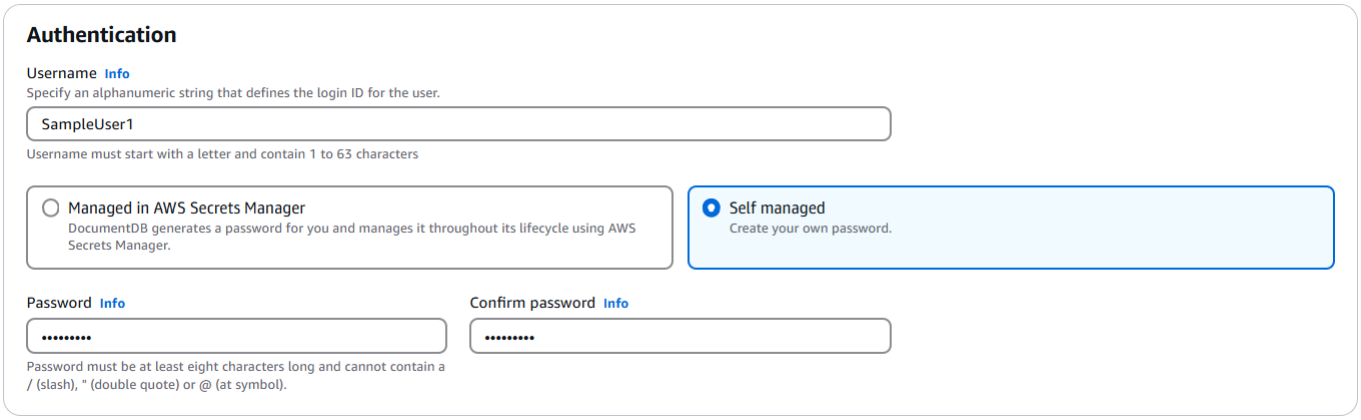


7. Behalten Sie im Abschnitt Konnektivität die Standardeinstellung „Keine Verbindung zu einer EC2 Rechenressource herstellen“ bei.



8. Geben Sie im Abschnitt Authentifizierung einen Benutzernamen für den Hauptbenutzer ein und wählen Sie dann Selbstverwaltet aus. Geben Sie ein Passwort ein und bestätigen Sie es.

   Wenn Sie stattdessen Verwaltet in wählen AWS Secrets Manager, finden Sie Passwortverwaltung mit Amazon DocumentDB und AWS Secrets Manager weitere Informationen unter.



9. Behalten Sie alle anderen Optionen als Standard bei und wählen Sie Create cluster aus.

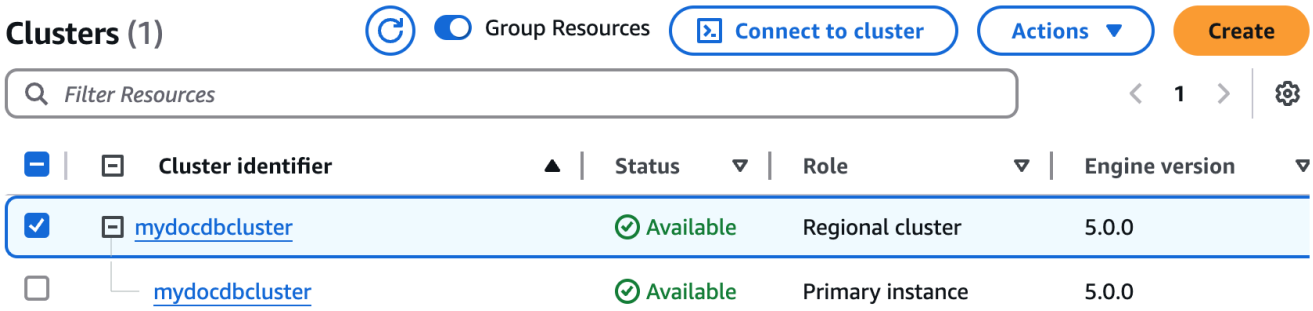
Amazon DocumentDB stellt jetzt Ihren Cluster bereit, was einige Minuten dauern kann.

> **Note**
>
> Informationen zu Cluster-Statuswerten finden Sie Werte für den Cluster-Status im Kapitel Monitoring Amazon DocumentDB.

Schritt 2: Connect zu Ihrem Cluster her

Stellen Sie mithilfe AWS CloudShell von Connect zu Ihrem Amazon DocumentDB-Cluster her.

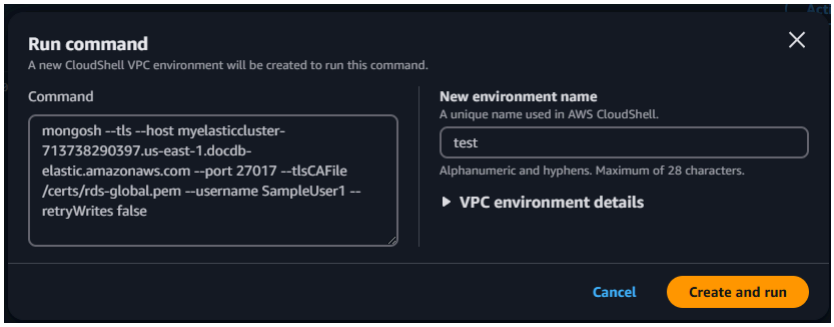
1. Suchen Sie auf der Amazon DocumentDB-Managementkonsole unter Clusters den Cluster, den Sie erstellt haben. Wählen Sie Ihren Cluster aus, indem Sie auf das Kontrollkästchen neben dem Cluster klicken.



2. Klicken Sie auf Mit Cluster verbinden (neben dem Dropdownmenü Aktionen). Diese Schaltfläche ist erst aktiviert, wenn Sie auf das Kontrollkästchen neben Ihrem Cluster geklickt haben und

der Status sowohl des regionalen Clusters als auch der primären Instance (s) als Verfügbar angezeigt wird. Der Befehlsbildschirm CloudShell Ausführen wird angezeigt.

3. Geben Sie im Feld Neuer Umgebungsname einen eindeutigen Namen ein, z. B. „Test“, und klicken Sie auf Erstellen und ausführen. VPC-Umgebungsdetails werden automatisch für Ihre Amazon DocumentDB DocumentDB-Datenbank konfiguriert.



4. Wenn Sie dazu aufgefordert werden, geben Sie das Passwort ein, das Sie in Schritt 1: Amazon DocumentDB-Cluster erstellen (Unterschritt 7) erstellt haben.



Nachdem Sie Ihr Passwort eingegeben haben und Ihre Aufforderung lautet`rs0 [direct: primary] <env-name>>`, sind Sie erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden.

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

Schritt 3: Daten einfügen und abfragen

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

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

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

   Sie erhalten die folgende Ausgabe:

   ```
   {
     acknowledged: true,
     insertedId: ObjectId('673657216bdf6258466b128c')
   }
   ```

2. Sie können das Dokument lesen, das Sie mit dem `findOne()` Befehl geschrieben haben (weil er nur ein einziges Dokument zurückgibt). Geben Sie Folgendes ein:

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

   Sie erhalten die folgende Ausgabe:

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

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

   ```
   db.profiles.insertMany([{ _id: 1, name: 'Matt', status: 'active', level: 12, score:
    202 },
         { _id: 2, name: 'Frank', status: 'inactive', level: 2, score: 9 },
         { _id: 3, name: 'Karen', status: 'active', level: 7, score: 87 },
         { _id: 4, name: 'Katie', status: 'active', level: 3, score: 27 }
   ])
   ```

   Sie erhalten die folgende Ausgabe:

   ```
   { acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3, '3': 4 } }
   ```

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

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

   Sie erhalten eine Ausgabe, die den Daten entspricht, die Sie in Schritt 3 eingegeben haben.

5. Verwenden Sie mithilfe eines Filters eine Abfrage für ein einzelnes Dokument. Geben Sie Folgendes ein:

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

   Sie erhalten die folgende Ausgabe:

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

6. Versuchen wir nun, ein Profil zu finden und es mit dem `findAndModify` Befehl zu ändern. Wir geben dem Benutzer Matt zusätzliche 10 Punkte mit dem folgenden Code:

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

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

```
{
    [{_id : 1, name : 'Matt', status: 'active', level: 12, score: 202}]
```

7. Sie können mit der folgenden Abfrage überprüfen, ob sich sein Punktestand geändert hat:

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

   Sie erhalten die folgende Ausgabe:

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

Schritt 4: Erkunden

Herzlichen Glückwunsch! Sie haben den Leitfaden Erste Schritte für Amazon DocumentDB DocumentDB-Instance-basierte Cluster erfolgreich abgeschlossen.

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

- Amazon DocumentDB verwalten
- Skalierung
- Sichern und Wiederherstellen

> **Note**
>
> Für den Cluster, den Sie im Rahmen dieser Übung „Erste Schritte“ erstellt haben, fallen weiterhin Kosten an, sofern Sie ihn nicht löschen. Anweisungen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Amazon DocumentDB DocumentDB-Schnellstart mit AWS CloudFormation

Dieser Abschnitt enthält Schritte und weitere Informationen, die Ihnen den schnellen Einstieg in die Nutzung von Amazon DocumentDB (mit MongoDB-Kompatibilität) erleichtern sollen. AWS CloudFormation Allgemeine Informationen zu Amazon DocumentDB finden Sie unterWas ist Amazon DocumentDB (mit MongoDB-Kompatibilität).

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

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

Dieser AWS CloudFormation Stapel ist nur für Tutorialzwecke vorgesehen. Wenn Sie diese Vorlage für eine Produktionsumgebung verwenden, empfehlen wir Ihnen, strengere IAM-Richtlinien und Sicherheitsvorkehrungen zu verwenden. Informationen zur Sicherung von Ressourcen finden Sie unter Amazon VPC Security und Amazon EC2 Network and Security.

Themen

- Voraussetzungen
- Einen Amazon DocumentDB DocumentDB-Stack AWS CloudFormation starten
- Zugreifen auf den Amazon DocumentDB-Cluster
- Kündigungsschutz und Löschschutz

Voraussetzungen

Bevor Sie einen Amazon DocumentDB-Cluster erstellen, müssen Sie über Folgendes verfügen:

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

Erforderliche IAM-Berechtigungen

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

AWS Verwaltete Richtlinien

- `AWSCloudFormationReadOnlyAccess`
- `AmazonDocDBFullAccess`

Zusätzliche IAM-Berechtigungen

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

Ersetzen Sie in den folgenden Beispielen jedes Beispiel *`user input placeholder`* durch die Informationen Ihrer Ressource.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocDBPermissions",
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBCluster",
                "rds:DeleteDBCluster",
                "rds:ModifyDBCluster",
                "rds:DescribeDBClusters",
                "rds:CreateDBInstance",
                "rds:DeleteDBInstance",
                "rds:ModifyDBInstance",
                "rds:DescribeDBInstances",
```

```
                "rds:CreateDBSubnetGroup",
                "rds:DeleteDBSecurityGroup",
                "rds:DescribeDBSubnetGroups"
            ],
            "Resource": [
                "arn:aws:rds:{AWS_REGION}:{AWS_ACCOUNT_ID}:cluster:*",
                "arn:aws:rds:{AWS_REGION}:{AWS_ACCOUNT_ID}:db:*",
                "arn:aws:rds:{AWS_REGION}:{AWS_ACCOUNT_ID}:subgrp:*"
            ]
        },
        {
            "Sid": "EC2NetworkingPermissions",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:DeleteSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets"
            ],
            "Resource": [
                "arn:aws:ec2:{AWS_REGION}:{AWS_ACCOUNT_ID}:security-group/*",
                "arn:aws:ec2:{AWS_REGION}:{AWS_ACCOUNT_ID}:vpc/*",
                "arn:aws:ec2:{AWS_REGION}:{AWS_ACCOUNT_ID}:subnet/*"
            ]
        },
        {
            "Sid": "EC2DescribePermissions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchLogsPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
```

```
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:{AWS_REGION}:{AWS_ACCOUNT_ID}:log-group:/aws/docdb/*",
                "arn:aws:logs:{AWS_REGION}:{AWS_ACCOUNT_ID}:log-group:/aws/docdb/*:log-
stream:*"
            ]
        },
        {
            "Sid": "KMSPermissions",
            "Effect": "Allow",
            "Action": [
                "kms:CreateKey",
                "kms:DescribeKey",
                "kms:EnableKey",
                "kms:ListKeys",
                "kms:PutKeyPolicy"
            ],
            "Resource": "arn:aws:kms:{AWS_REGION}:{AWS_ACCOUNT_ID}:key/*"
        },
        {
            "Sid": "KMSEncryption",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:{AWS_REGION}:{AWS_ACCOUNT_ID}:key/*",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "rds.{AWS_REGION}.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "IAMServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::{AWS_ACCOUNT_ID}:role/aws-service-role/
rds.amazonaws.com/AWSServiceRoleForRDS",
            "Condition": {
```

```
                "StringEquals": {
                    "iam:AWSServiceName": "rds.amazonaws.com"
                }
            }
        }
    ]
}
```

EC2 Amazon-Schlüsselpaar

In der Region, in der Sie den AWS CloudFormation Stack erstellen möchten, muss ein key pair (und die PEM-Datei) verfügbar sein. Wenn Sie ein key pair erstellen müssen, finden Sie weitere Informationen unter Erstellen eines Schlüsselpaars mithilfe von Amazon EC2 im EC2 Amazon-Benutzerhandbuch.

Einen Amazon DocumentDB DocumentDB-Stack AWS CloudFormation starten

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

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

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

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

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

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

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

1. Wählen Sie in der unteren rechten Ecke der Seite „Stack erstellen“ die Option In Designer anzeigen aus.
2. Ändern Sie die Vorlage mithilfe des integrierten JSON- und YAML-Editors auf der resultierenden AWS CloudFormation Designer-Seite der Konsole. Blättern Sie zum Abschnitt `Resources` und ändern Sie diesen wie folgt, um den `DeletionProtection` einzubinden. Weitere Informationen zur Verwendung von AWS CloudFormation Designer finden Sie unter Was ist AWS CloudFormation Designer? .

   JSON:

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

   YAML:

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

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

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

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

       Einschränkungen für Instance-Benennungen:
       - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
       - Muss mit einem Buchstaben beginnen.

- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- Muss für alle Instances in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.

- MasterPassword— Das Passwort für das Datenbank-Administratorkonto.
- MasterUser— Der Benutzername des Datenbank-Administratorkontos. Der MasterUser muss mit einem Buchstaben beginnen und darf nur alphanumerische Zeichen enthalten.

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

5. Stack-Optionen konfigurieren — Konfigurieren Sie die Tags, Berechtigungen und zusätzlichen Optionen Ihres Stacks.

- Tags — Geben Sie Tag-Paare (Schlüssel-Wert) an, die auf Ihre Ressourcen in Ihrem Stack angewendet werden sollen. Sie können bis zu 50 eindeutige Tags für jeden Stack hinzufügen.
- Berechtigungen — optional. Wählen Sie eine IAM-Rolle, um explizit zu definieren, wie Ressourcen im Stack erstellt, geändert oder gelöscht werden AWS CloudFormation können. Wenn Sie keine Rolle auswählen, AWS CloudFormation verwendet Berechtigungen, die auf Ihren Benutzeranmeldedaten basieren. Bevor Sie eine Service-Rolle festlegen, stellen Sie sicher, dass Sie über die Berechtigung zum Weiterleiten (`iam:PassRole`) verfügen. Die `iam:PassRole`-Berechtigung gibt an, welche Rollen Sie verwenden können.

  > **Note**
  >
  > Wenn Sie eine Servicerolle angeben, verwendet diese Rolle AWS CloudFormation immer für alle Operationen, die auf diesem Stack ausgeführt werden. Andere Benutzer, die Berechtigungen zum Ausführen von Vorgängen in diesem Stapel haben, können diese Rolle verwenden, selbst wenn Sie keine Berechtigung zum Weiterleiten haben. Wenn die Rolle Berechtigungen umfasst, die der Benutzer nicht haben sollte, können Sie die Berechtigungen eines Benutzers versehentlich weiterleiten. Stellen Sie sicher, dass die Rolle die geringsten Rechte gewährt.

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

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

Sie können die Stack-Richtlinie direkt im JSON-Format eingeben oder eine JSON-Datei, die die Stack-Richtlinie enthält, hochladen. Weitere Informationen finden Sie unter Prevent Updates to Stack Resources (Verhindern von Aktualisierungen der Stack-Ressourcen).

- Rollback-Konfiguration — Optional. Geben Sie CloudWatch Log-Alarme AWS CloudFormation an, die bei der Erstellung und Aktualisierung des Stacks überwacht werden sollen. Wenn der Vorgang einen Alarmschwellenwert überschreitet, wird er AWS CloudFormation rückgängig gemacht.
- Benachrichtigungsoptionen — optional. Geben Sie Themen für das Simple Notification System (SNS) an.
- Optionen zur Stapelerstellung — Optional. Sie können die folgenden Optionen angeben:
  - Rollback bei einem Fehler — Gibt an, ob der Stack zurückgesetzt werden soll, falls die Stack-Erstellung fehlschlägt.
  - Timeout — Die Anzahl der Minuten, bevor bei der Stack-Erstellung ein Timeout eintritt.
  - Kündigungsschutz — verhindert, dass der Stack versehentlich gelöscht wird.

> **Note**
>
> AWS CloudFormation Der Kündigungsschutz unterscheidet sich vom Löschschutzkonzept von Amazon DocumentDB. Weitere Informationen finden Sie unter Kündigungsschutz und Löschschutz.

Wählen Sie Next (Weiter), um fortzufahren.

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

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

Zugreifen auf den Amazon DocumentDB-Cluster

Sobald der AWS CloudFormation Stack abgeschlossen ist, können Sie eine EC2 Amazon-Instance verwenden, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen. Informationen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance mithilfe von SSH finden Sie unter Connect to Your Linux Instance im EC2 Amazon-Benutzerhandbuch.

Nachdem Sie die Verbindung hergestellt haben, lesen Sie die folgenden Abschnitte, die Informationen zur Verwendung von Amazon DocumentDB enthalten.

- Schritt 3: Daten einfügen und abfragen
- Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen
- Überwachung von Amazon DocumentDB

Kündigungsschutz und Löschschutz

Es ist eine bewährte Methode von Amazon DocumentDB, den Löschschutz und den Kündigungsschutz zu aktivieren. CloudFormation Der Kündigungsschutz unterscheidet sich deutlich von der Amazon DocumentDB DocumentDB-Löschschutzfunktion.

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

Amazon DocumentDB DocumentDB-Kompatibilität mit MongoDB

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

Themen

- MongoDB 5.0-Kompatibilität
- MongoDB 4.0-Kompatibilität

MongoDB 5.0-Kompatibilität

Themen

- Was ist neu in Amazon DocumentDB 5.0
- Erste Schritte mit Amazon DocumentDB 5.0
- Upgrade oder Migration auf Amazon DocumentDB 5.0
- Funktionelle Unterschiede

Was ist neu in Amazon DocumentDB 5.0

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

- Das Speicherlimit wurde für alle instanzbasierten Amazon DocumentDB-Cluster und shard-basierten Elastic-Cluster auf 128 TiB erhöht.

- Amazon DocumentDB 5.0 Engine (Version 3.0.775) eingeführt
  - Support für MongoDB 5.0 API-Treiber
  - Support für clientseitige Verschlüsselung auf Feldebene (FLE). Sie können jetzt Felder auf der Clientseite verschlüsseln, bevor Sie die Daten in den Amazon DocumentDB-Cluster schreiben. Weitere Informationen finden Sie unter Clientseitige Verschlüsselung auf Feldebene.
  - Neue Aggregationsoperatoren:, `$dateAdd` `$dateSubtract`
  - Unterstützt Indizes mit `$elemMatch` Operator. Das hat zur Folge, `$elemMatch` dass Abfragen zu Indexscans führen.

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

Erste Schritte mit Amazon DocumentDB 5.0

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

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

Amazon DocumentDB 5.0 unterstützt Amazon EC2 Graviton2-Prozessoren wie `r6g` `t4.medium` Instance-Typen für Ihre Cluster und ist in allen unterstützten Regionen verfügbar. Weitere Informationen zur Preisgestaltung finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) — Preise.

Upgrade oder Migration auf Amazon DocumentDB 5.0

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

Funktionelle Unterschiede

Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und 5.0

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

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

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

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

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

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

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

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

MongoDB 4.0-Kompatibilität

Themen

- Funktionen von Amazon DocumentDB 4.0
- Erste Schritte mit Amazon DocumentDB 4.0
- Upgrade oder Migration zu Amazon DocumentDB 4.0
- Funktionelle Unterschiede

Funktionen von Amazon DocumentDB 4.0

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

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

Abfrage), die Möglichkeit`findAndModify`, `hint()` mit, Leistungsoptimierungen für `$addToSet` und Verbesserungen zur Reduzierung der Gesamtindexgröße. Weitere Informationen finden Sie unter Versionshinweise.

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

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

Erste Schritte mit Amazon DocumentDB 4.0

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

> **Note**
>
> Wenn Sie das AWS SDK oder AWS CloudFormation verwenden AWS CLI, wird die Engine-Version standardmäßig auf 5.0.0 gesetzt. Sie müssen den Parameter explizit angeben`engineVersion = 4.0.0`, um einen neuen Amazon DocumentDB 4.0-Cluster

oder einen neuen Amazon DocumentDB 3.6-Cluster `engineVersion = 3.6.0` zu erstellen. Für einen bestimmten Amazon DocumentDB-Cluster können Sie die Cluster-Version ermitteln, indem Sie AWS CLI den aufrufen `describe-db-clusters` oder die Amazon DocumentDB-Managementkonsole verwenden, um die Engine-Versionsnummer für einen bestimmten Cluster anzuzeigen.

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

Upgrade oder Migration zu Amazon DocumentDB 4.0

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

Funktionelle Unterschiede

Funktionale Unterschiede zwischen Amazon DocumentDB 3.6 und 4.0

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

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

`{"b":1}`) bietet jetzt eine konsistente Ausgabe für alle numerischen Datentypen und für jede Ebene eines Dokuments.

Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und MongoDB 4.0

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

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

Transaktionen in Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) unterstützt jetzt MongoDB 4.0-Kompatibilität, einschließlich Transaktionen. Sie können Transaktionen für mehrere Dokumente, Kontoauszüge, Sammlungen und Datenbanken durchführen. Transaktionen vereinfachen die Anwendungsentwicklung, indem sie es Ihnen ermöglichen, atomare, konsistente, isolierte und dauerhafte Operationen (ACID) für ein oder mehrere Dokumente innerhalb eines Amazon DocumentDB-Clusters durchzuführen. Zu den häufigsten Anwendungsfällen für Transaktionen gehören die Finanzabwicklung, die Ausführung und Verwaltung von Bestellungen sowie die Entwicklung von Spielen für mehrere Spieler.

Für Transaktionen fallen keine zusätzlichen Kosten an. Sie zahlen nur für die Lese- und Schreibvorgänge IOs , die Sie im Rahmen der Transaktionen verbrauchen.

Themen

- Voraussetzungen
- Bewährte Methoden
- Einschränkungen
- Überwachung und Diagnose
- Isolationsstufe für Transaktionen
- Anwendungsfälle
- Unterstützte Befehle
- Nicht unterstützte Funktionen
- Sitzungen
- Transaktionsfehler

Voraussetzungen

Um die Transaktionsfunktion nutzen zu können, müssen Sie die folgenden Anforderungen erfüllen:

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

Bewährte Methoden

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

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

Einschränkungen

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

- Transaktionen haben ein Ausführungslimit von einer Minute und Sitzungen haben ein 30-minütiges Timeout. Wenn bei einer Transaktion das Timeout überschritten wird, wird sie abgebrochen, und alle nachfolgenden Befehle, die innerhalb der Sitzung für die bestehende Transaktion ausgegeben werden, führen zu dem folgenden Fehler:

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

Überwachung und Diagnose

Mit der Unterstützung für Transaktionen in Amazon DocumentDB 4.0 wurden zusätzliche CloudWatch Metriken hinzugefügt, die Ihnen helfen, Ihre Transaktionen zu überwachen.

Neue Metriken CloudWatch

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

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

Darüber hinaus wurden neue Felder zu `currentOp` `lsidtransactionThreadId`, und ein neuer Status für „`idle transaction`“ und `serverStatus` Transaktionen hinzugefügt:`currentActive`,`currentInactive`,`currentOpen`, `totalAbortedtotalCommitted`, und`totalStarted`.

Isolationsstufe für Transaktionen

Wenn Sie eine Transaktion starten, können Sie sowohl als auch angeben, `writeConcern` wie im folgenden Beispiel gezeigt: `readConcern`

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

Denn `readConcern` Amazon DocumentDB unterstützt standardmäßig die Snapshot-Isolierung. Wenn „Lokal“, „Verfügbar“ oder „Mehrheit“ angegeben ist, führt Amazon DocumentDB ein Upgrade der `readConcern` Stufe auf Snapshot durch. `readConcern` Amazon DocumentDB unterstützt das Linearisierbare nicht `readConcern` und die Angabe eines solchen Leseproblems führt zu einem Fehler.

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

Anwendungsfälle

In diesem Abschnitt werden wir uns mit zwei Anwendungsfällen für Transaktionen befassen: Multi-Statement und Multi-Collection.

Transaktionen mit mehreren Kontoauszügen

Amazon DocumentDB-Transaktionen bestehen aus mehreren Anweisungen, was bedeutet, dass Sie eine Transaktion schreiben können, die sich über mehrere Anweisungen mit einem expliziten Commit oder Rollback erstreckt. Sie können `findAndModify` Aktionen`insert`, `update``delete`, und als eine einzige atomare Operation gruppieren.

Ein häufiger Anwendungsfall für Transaktionen mit mehreren Kontoauszügen ist eine Debit-/Kredittransaktion. Zum Beispiel: Sie schulden einem Freund Geld für Kleidung. Sie müssen also 500 USD von Ihrem Konto abbuchen (abheben) und dem Konto Ihres Freundes 500 USD (Einzahlung) gutschreiben. Um diesen Vorgang durchzuführen, führen Sie sowohl den Debit- als auch den Kreditvorgang innerhalb einer einzigen Transaktion durch, um die Atomarität sicherzustellen. Dadurch werden Szenarien vermieden, in denen 500$ von Ihrem Konto abgebucht, aber nicht dem Konto Ihres Freundes gutgeschrieben werden. So würde dieser Anwendungsfall aussehen:

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

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

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

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

session.startTransaction();

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

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

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

session.commitTransaction();

accountColl.find();

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

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

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

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

session.startTransaction();

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

session.abortTransaction();
```

Transaktionen mit mehreren Sammelaktionen

Bei unseren Transaktionen handelt es sich auch um Sammeltransaktionen, was bedeutet, dass sie für mehrere Operationen innerhalb einer einzigen Transaktion und für mehrere Inkassovorgänge verwendet werden können. Dies ermöglicht eine konsistente Ansicht der Daten und gewährleistet die Integrität Ihrer Daten. Wenn Sie die Befehle einzeln übergeben<>, handelt es sich bei den Transaktionen um all-or-nothing Ausführungen, d. h. sie werden entweder alle erfolgreich ausgeführt oder alle schlagen fehl.

Hier ist ein Beispiel für Transaktionen mit mehreren Sammlungen, wobei dasselbe Szenario und die Daten aus dem Beispiel für Transaktionen mit mehreren Kontoauszügen verwendet werden.

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

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

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

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

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

session.startTransaction();

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

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

session.commitTransaction();

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

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

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

```
var amountToTransfer = 500;

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

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

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

session.startTransaction();

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

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

session.abortTransaction();

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

Beispiele für Transaktions-APIs für die Callback-API

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

Javascript

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Javascript verwendet wird.

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

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

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

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

session.startTransaction();

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

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

session.commitTransaction();

accountColl.find();
```

Node.js

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

```
// Node.js callback API:

const bankDB = await mongoclient.db("bank");
var accountColl = await bankDB.createCollection("account");
var amountToTransfer = 500;
```

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

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

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

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

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

C#

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

```
// C# Callback API

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

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

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

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

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

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

Ruby

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Ruby verwendet wird.

```
// Ruby Callback API

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

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

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

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

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

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

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

session.end_session
```

Go

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

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

ctx := context.TODO()

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

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

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

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

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

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

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

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

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

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

Java

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Java verwendet wird.

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

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

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

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

            // check Alice's new balance
            documentList = new ArrayList<>();
```

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

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

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

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

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

C

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit C verwendet wird.

```
// Sample Code for C with Callback

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

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

```
} ctx_t;

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

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

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

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

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

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

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

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

```
   mongoc_client_session_t *client_session = mongoc_client_start_session (client,
session_opts, &error);

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

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

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

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

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

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

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

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

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

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

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

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

    // cleanup
    bson_destroy(alice_query);
    bson_destroy(bob_query);
    mongoc_client_session_destroy(client_session);
    bson_destroy(opts);
    mongoc_transaction_opts_destroy(txn_opts);
    mongoc_read_concern_destroy(read_concern);
    mongoc_write_concern_destroy(write_concern);
    mongoc_cursor_destroy(cursor);
    bson_destroy(doc);
}
int main(int argc, char* argv[]) {
```

```
    mongoc_init ();
    mongoc_client_t* client = mongoc_client_new (<connection uri>);
    bson_error_t error;

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

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

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

    bool r = true;

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

    test_callback_money_transfer(client, collection, amount_to_transfer);

}
```

Python

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Python verwendet wird.

```
// Sample Python code with callback api

import pymongo
```

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

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

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

amount_to_transfer = 500

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

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

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

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

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

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

Beispiele für Transaktions-APIs für die Kern-API

Javascript

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Javascript verwendet wird.

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

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

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

session.startTransaction();

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

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

session.commitTransaction();

accountColl.find();
```

C#

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

```
// C# Core API

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

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

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

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

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

    session.CommitTransaction();
 }

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

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

```
                    }
                    else
                    {
                        throw;
                    }
                }
            }

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

Ruby

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Ruby verwendet wird.

```
# Ruby Core API

def transfer_money_w_retry(session, accountColl)
    amountToTransfer = 500

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

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

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

    session.commit_transaction

end

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

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

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

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

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

end
```

Go

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

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

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

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

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

    var result Account
    // deduct $500 from Alice's account
    err := accountColl.FindOne(sessionContext, bson.M{"name":
 "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    newAliceBalance := aliceBalance - amountToTransfer
    _, err = accountColl.UpdateOne(sessionContext, bson.M{"name": "Alice"},
 bson.M{"$set": bson.M{"balance": newAliceBalance}})
    if err != nil {
        sessionContext.AbortTransaction(sessionContext)
    }
    err = accountColl.FindOne(sessionContext, bson.M{"name":
 "Alice"}).Decode(&result)
    aliceBalance = result.Balance
```

```
    assert.Equal(t, aliceBalance, newAliceBalance)

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

    err = sessionContext.CommitTransaction(sessionContext)
    return err
}

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

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

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

```
            println("transaction failed")
            return err
        }
    })

    // check results outside of transaction
    var result Account
    accountColl.FindOne(ctx, bson.M{"name": "Alice"}).Decode(&result)
    aliceBalance := result.Balance
    assert.Equal(t, aliceBalance, 500)
    accountColl.FindOne(ctx, bson.M{"name": "Bob"}).Decode(&result)
    bobBalance := result.Balance
    assert.Equal(t, bobBalance, 1500)
}
```

Java

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Java verwendet wird.

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

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

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

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

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

```
                throw e;
            }
        }
    }
}

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

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

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

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

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

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

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

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

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

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

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

```
            }
        }
    }
}

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

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

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

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

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

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

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

    // add $500 to Bob's account
    findCallback = new SubscriberLatchWrapper<>();
    accountColl.find(session, new Document("name",
 "Bob")).first().subscribe(findCallback);
    documentFound = findCallback.get().get(0);
    int bobBalance = (int) documentFound.get("balance");
    int newBobBalance = bobBalance + amountToTransfer;

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

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

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

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

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

```
    /**
     * Construct an instance
     */
    public SubscriberLatchWrapper() {
        this.received = new ArrayList<>();
        this.errors = new ArrayList<>();
        this.latch = new CountDownLatch(1);
    }

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

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

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

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

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

```
    }

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

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

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

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

C

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit C verwendet wird.

```
// Sample C code with core session

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

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

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

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

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

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

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

    bson_t reply;
    bool r = true;
    const bson_t *doc;
```

```
    bson_iter_t iter;
    bson_error_t error;

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

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

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

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

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

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

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

```
    assert(alice_balance == 500);

    // add 500 to Bob's balance
    // find account balance of Bob
    cursor = mongoc_collection_find_with_opts (collection, bob_query, NULL, NULL);
    mongoc_cursor_next (cursor, &doc);
    bson_iter_init (&iter, doc);
    bson_iter_find (&iter, "balance");
    int64_t bob_balance = (bson_iter_value (&iter))->value.v_int64;
    int64_t new_bob_balance = bob_balance + amount_to_transfer;

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

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

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

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

    // connect to bank db
    mongoc_database_t *database = mongoc_client_get_database (client, "bank");
    // access account collection
    mongoc_collection_t* collection = mongoc_client_get_collection(client, "bank",
 "account");
    // set amount to transfer
```

```
    int64_t amount_to_transfer = 500;
    // delete the collection if already existing
    mongoc_collection_drop(collection, &error);

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

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

    bool r = true;

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

    test_core_money_transfer(client, collection, amount_to_transfer);

}
```

Scala

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Scala verwendet wird.

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

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

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

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

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

    clientSession
    })

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

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

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

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

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

        }
        }
    }

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

    accountColl.drop().await()

}
```

Python

Der folgende Code zeigt, wie die Amazon DocumentDB-Transaktions-API mit Python verwendet wird.

```
// Sample Python code with Core api

import pymongo

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

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

```
collection.insert_one({"_id": 2, "name": "Bob", "balance": 1000})

amount_to_transfer = 500

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

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

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

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

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

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

Unterstützte Befehle

Befehl	Unterstützt
`abortTransaction`	Ja
`commitTransaction`	Ja
`endSessions`	Ja

Befehl	Unterstützt
`killSession`	Ja
`killAllSession`	Ja
`killAllSessionsByPattern`	Nein
`refreshSessions`	Nein
`startSession`	Ja

Nicht unterstützte Funktionen

Methoden	Stufen oder Befehle

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

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

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

Sitzungen

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

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

Bei der Verwendung von Transaktionen innerhalb von Amazon DocumentDB wird eine Transaktion innerhalb einer Sitzung mithilfe der `session.startTransaction()` API initiiert, und eine Sitzung unterstützt jeweils eine einzelne Transaktion. In ähnlicher Weise werden Transaktionen entweder mit Commit (`session.commitTransaction()`) oder Abort (`session.abortTransaction()`) abgeschlossen. APIs

Kausale Konsistenz

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

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

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

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

```
mySession.endSession()
```

Sie können die kausale Konsistenz innerhalb einer Sitzung deaktivieren. Bitte beachten Sie, dass Sie auf diese Weise das Session-Framework nutzen können, aber keine Garantie für kausale Konsistenz bei Lesevorgängen bietet. Bei der Verwendung von Amazon DocumentDB sind die Lesevorgänge von der Primärinstanz read-after-write konsistent und die Lesevorgänge von den Replikatinstanzen werden letztendlich konsistent sein. Transaktionen sind der Hauptanwendungsfall für die Nutzung von Sitzungen.

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

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

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

Wiederholbare Schreibvorgänge

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

> **Note**
>
> Wenn Sie eine ältere Mongo-Shell (nicht Mongosh) verwenden, fügen Sie den `retryWrites=false` Befehl in keine Codezeichenfolge ein. Standardmäßig sind wiederholbare Schreibvorgänge deaktiviert. `retryWrites=false`Das Einschließen kann zu einem Fehler bei normalen Lesebefehlen führen.

Transaktionsfehler

Bei der Verwendung von Transaktionen gibt es Szenarien, in denen ein Fehler auftreten kann, der besagt, dass eine Transaktionsnummer mit keiner laufenden Transaktion übereinstimmt.

Der Fehler kann in mindestens zwei verschiedenen Szenarien generiert werden:

- Nach dem Transaktions-Timeout von einer Minute.
- Nach einem Neustart der Instanz (aufgrund von Patches, Wiederherstellung nach einem Absturz usw.) kann dieser Fehler auch dann auftreten, wenn die Transaktion erfolgreich festgeschrieben wurde. Während eines Instance-Neustarts kann die Datenbank den Unterschied zwischen einer erfolgreich abgeschlossenen Transaktion und einer abgebrochenen Transaktion nicht erkennen. Mit anderen Worten, der Status des Transaktionsabschlusses ist mehrdeutig.

Der beste Weg, diesen Fehler zu beheben, besteht darin, Transaktionsaktualisierungen idempotent zu machen — zum Beispiel, indem Sie den `$set` Mutator anstelle einer Inkrement-/Dekrementierungsoperation verwenden. Siehe unten:

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

Bewährte Methoden für Amazon DocumentDB

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

Themen

- Grundlegende Anleitungen für den Betrieb
- Dimensionierung der Instanzen
- Arbeiten mit Indizes
- Bewährte Methoden für die Gewährleistung der Sicherheit
- Kostenoptimierung
- Verwendung von Metriken zur Identifizierung von Problemen mit der Leistung
- TTL- und Zeitreihen-Workloads
- Migrationen
- Arbeiten mit Cluster-Parametergruppen
- Abfragen in der Aggregationspipeline
- batchInsert und batchUpdate

Grundlegende Anleitungen für den Betrieb

Im Folgenden finden Sie grundlegende Betriebsrichtlinien, die jeder bei der Arbeit mit Amazon DocumentDB beachten sollte. Das Amazon DocumentDB Service Level Agreement verlangt, dass Sie diese Richtlinien befolgen.

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

Amazon so ein, dass Sie benachrichtigt werden, wenn sich die Nutzungsmuster ändern oder wenn Sie die Kapazität Ihrer Bereitstellung fast erreicht haben.

- Skalieren Sie Ihre Instances, wenn Sie die Grenzen der Speicherkapazität beinahe erreicht haben. Ihre Instances sollten mit genügend Rechenressourcen (d. h. RAM, CPU) ausgestattet sein, um Nachfragesteigerungen seitens Ihrer Anwendungen bewältigen zu können.
- Legen Sie die Aufbewahrungszeitraum für Backups so fest, dass sie zu Ihrem Wiederherstellungsziel passt.
- Testen Sie den Failover für Ihren Cluster, um zu verstehen, wie lange der Vorgang für Ihren Anwendungsfall dauert. Weitere Informationen finden Sie unter Amazon DocumentDB-Failover.
- Stellen Sie mit dem Cluster-Endpunkt (siheAmazon DocumentDB DocumentDB-Endpunkte) und im Replikatsatzmodus (siheAls Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen) eine Connect zu Ihrem Amazon DocumentDB-Cluster her, um die Auswirkungen eines Failovers auf Ihre Anwendung zu minimieren.
- Wählen Sie eine Treibereinstellung für die Leseeinstellung aus, die die Leseskalierung maximiert und gleichzeitig die Anforderungen für die Lesekonsistenz Ihrer Anwendung erfüllt. Die Leseeinstellung `secondaryPreferred` ermöglicht Replica-Lesevorgänge, sodass die primäre Instance produktiver sein kann. Weitere Informationen finden Sie unter Lesen Sie die Einstellungsoptionen.
- Entwerfen Sie Ihre Anwendung so, dass Sie im Falle von Netzwerk- und Datenbankfehlern stabil ist. Unterscheiden Sie mithilfe des Fehlermechanismus des Treibers zwischen vorübergehenden Fehlern und persistenten Fehlern. Wiederholen Sie den Vorgang bei vorübergehenden Fehlern mit einem exponentiellen Backoff-Mechanismus (bei Bedarf). Stellen Sie sicher, dass Ihre Anwendung bei der Implementierung von Logik für Wiederholversuche die Datenkonsistenz berücksichtigt.
- Aktivieren Sie den Cluster-Löschschutz für alle Produktions-Cluster oder Cluster mit wertvollen Daten. Bevor Sie einen Amazon DocumentDB-Cluster löschen, erstellen Sie einen letzten Snapshot. Wenn Sie Ressourcen mit bereitstellen AWS CloudFormation, aktivieren Sie den Kündigungsschutz. Weitere Informationen finden Sie unter Kündigungsschutz und Löschschutz.
- Bei der Erstellung eines Amazon DocumentDB-Clusters `--engine-version` ist dies ein optionaler Parameter, der standardmäßig die neueste Hauptversion der Engine verwendet. Die aktuelle Version der Haupt-Engine ist 5.0.0. Wenn neue Hauptversionen der Engine veröffentlicht werden, `--engine-version` wird die Standard-Engine-Version für aktualisiert, sodass sie die letzte Hauptversion der Engine wiedergibt. Daher empfehlen wir für Produktionsworkloads, insbesondere solche, die von Skripten, Automatisierung oder AWS CloudFormation Vorlagen abhängig sind, die Angabe der `--engine-version` beabsichtigten Hauptversion ausdrücklich.

Dimensionierung der Instanzen

Einer der wichtigsten Aspekte bei der Auswahl einer Instance-Größe in Amazon DocumentDB ist die Größe des Arbeitsspeichers für Ihren Cache. Amazon DocumentDB reserviert ein Drittel des RAM für seine eigenen Dienste, was bedeutet, dass nur zwei Drittel des Instance-RAM für den Cache verfügbar sind. Daher ist es eine bewährte Methode von Amazon DocumentDB, einen Instance-Typ auszuwählen, der über ausreichend RAM verfügt, damit Ihr Arbeitssatz (d. h. Daten und Indizes) im Arbeitsspeicher Platz findet. Instances mit der richtigen Größe können Sie die Gesamtleistung optimieren und möglicherweise die Kosten minimieren I/O .

Um festzustellen, ob das Working-Set Ihrer Anwendung in den Arbeitsspeicher passt, überwachen `BufferCacheHitRatio` Sie die Nutzung von Amazon CloudWatch für jede Instance in einem Cluster, die unter Last steht.

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

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

Abhängig von Ihren Workload- und Latenzanforderungen kann es akzeptabel sein, dass Ihre Anwendung während der stabilen Zustandsnutzung höhere `BufferCacheHitRatio`-Werte aufweist, das `BufferCacheHitRatio` jedoch periodisch nachlässt, da analytische Abfragen,

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

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

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

Arbeiten mit Indizes

Erstellen von Indizes

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

Index-Selektivität

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

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

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

Einfluss von Indizes auf das Schreiben von Daten

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

Cluster-Instances müssen über eine angemessene Größe verfügen, um den gesamten Arbeitsspeicher zu erhalten. Dadurch wird vermieden, dass Indexseiten kontinuierlich vom Speichervolumen gelesen werden müssen, was sich negativ auf die Leistung auswirkt und höhere I/O Kosten verursacht. Weitere Informationen finden Sie unter Dimensionierung der Instanzen.

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

Identifizierung fehlender Indizes

Die Identifizierung fehlender Indizes ist eine bewährte Methode, die wir empfehlen, regelmäßig durchzuführen. Weitere Informationen finden Sie unter Wie identifiziere ich fehlende Indizes?.

Identifizieren ungenutzter Indizes

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

Bewährte Methoden für die Gewährleistung der Sicherheit

Aus Sicherheitsgründen müssen Sie AWS Identity and Access Management (IAM-) Konten verwenden, um den Zugriff auf Amazon DocumentDB DocumentDB-API-Operationen zu kontrollieren, insbesondere Operationen, die Amazon DocumentDB DocumentDB-Ressourcen erstellen, ändern oder löschen. Zu solchen Ressourcen gehören Cluster, Sicherheitsgruppen und Parametergruppen. Sie müssen IAM auch zur Steuerung von Aktionen verwenden, mit denen allgemeine Verwaltungsaktionen wie das Sichern und Wiederherstellen von Clustern ausgeführt werden. Verwenden Sie bei der Erstellung von IAM-Rollen das Prinzip der geringsten Rechte.

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

- Wechseln Sie regelmäßig die IAM-Anmeldeinformationen.
- Konfigurieren Sie AWS Secrets Manager so, dass die Secrets für Amazon DocumentDB automatisch rotiert werden. Weitere Informationen finden Sie unter Rotating Your AWS Secrets Manager Secrets und Rotating Secrets for Amazon DocumentDB im AWS Secrets Manager User Guide.
- Erteilen Sie jedem Amazon DocumentDB DocumentDB-Benutzer die Mindestberechtigungen, die zur Erfüllung seiner Aufgaben erforderlich sind. Weitere Informationen finden Sie unter Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle.
- Verwenden Sie Transport Layer Security (TLS), um Ihre Daten bei der Übertragung und AWS KMS um Ihre Daten im Ruhezustand zu verschlüsseln.

Kostenoptimierung

Die folgenden bewährten Methoden können Ihnen helfen, Ihre Kosten bei der Verwendung von Amazon DocumentDB zu verwalten und zu minimieren. Preisinformationen finden Sie unter Amazon DocumentDB (mit MongoDB-Kompatibilität) und Amazon DocumentDB (mit MongoDB-Kompatibilität). FAQs

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

Verwendung von Metriken zur Identifizierung von Problemen mit der Leistung

Themen

- Anzeigen von Leistungsmetriken
- Einen CloudWatch Alarm einrichten
- Auswerten von Leistungsmetriken
- Auswertung der Nutzung von Amazon DocumentDB DocumentDB-Instances anhand von Metriken CloudWatch
- Optimieren von Abfragen

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

Anzeigen von Leistungsmetriken

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

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

Sie können Leistungskennzahlen mit dem AWS Management Console oder anzeigen. AWS CLI Weitere Informationen finden Sie unter CloudWatch Daten anzeigen.

Einen CloudWatch Alarm einrichten

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

Auswerten von Leistungsmetriken

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

CPU

- CPU-Auslastung — Der Prozentsatz der verwendeten Computerverarbeitungskapazität.

Arbeitsspeicher

- Freier Arbeitsspeicher — Wie viel RAM ist auf der Instance verfügbar.
- Swap-Nutzung — Wie viel Swap-Speicherplatz von der Instance verwendet wird, in Megabyte.

Eingabe-/Ausgabe-Operationen

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

Netzwerkdatenverkehr

- Netzwerk-Empfangsdurchsatz, Netzwerkübertragungsdurchsatz — Die Rate des Netzwerkverkehrs zur und von der Instance in Megabyte pro Sekunde.

Datenbankverbindungen

- DB-Verbindungen — Die Anzahl der Client-Sitzungen, die mit der Instance verbunden sind.

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

Nachstehend folgen Empfehlungen und Ratschläge zu bestimmten Arten von Metriken:

- Hoher CPU-Verbrauch — Hohe Werte für die CPU-Auslastung können angemessen sein, vorausgesetzt, sie entsprechen Ihren Zielen für Ihre Anwendung (wie Durchsatz oder Parallelität) und werden erwartet. Wenn die CPU-Auslastung konsistent mehr als 80 Prozent beträgt, sollten Sie eine Aufwärtsskalierung Ihrer Instances in Betracht ziehen.
- Hoher RAM-Verbrauch — Wenn Ihre `FreeableMemory` Kennzahl häufig unter 10% des gesamten Instance-Speichers fällt, sollten Sie eine Skalierung Ihrer Instances in Betracht ziehen. Weitere Informationen darüber, was passiert, wenn Ihre DocumentDB-Instance einer hohen Speicherauslastung ausgesetzt ist, finden Sie unter Amazon DocumentDB Resource Governance.
- Swap-Nutzung — Diese Metrik sollte bei oder nahe Null bleiben. Wenn die Swap-Nutzung erheblich ist, sollten Sie eine Aufwärtsskalierung Ihrer Instances in Betracht ziehen.
- Netzwerkverkehr — Wenden Sie sich bezüglich des Netzwerkverkehrs an Ihren Systemadministrator, um zu erfahren, wie hoch der zu erwartende Durchsatz für Ihr Domänennetzwerk und Ihre Internetverbindung ist. Überprüfen Sie den Netzwerkdatenverkehr, wenn der Durchsatz dauerhaft unter dem erwarteten Wert liegt.
- Datenbankverbindungen — Erwägen Sie, Datenbankverbindungen einzuschränken, wenn Sie eine hohe Anzahl von Benutzerverbindungen zusammen mit einer Verringerung der Instanzleistung und Reaktionszeit feststellen. Die optimale Anzahl der Benutzerverbindungen für Ihre Instance ist von der Instance-Klasse und der Komplexität der ausgeführten Operationen abhängig. Bei Problemen mit Leistungsmetriken sollten Sie zunächst die am häufigsten verwendeten und kostspieligsten Abfragen anpassen, um festzustellen, ob dies den Druck auf die Systemressourcen verringert und die Leistung verbessert.

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

Auswertung der Nutzung von Amazon DocumentDB DocumentDB-Instances anhand von Metriken CloudWatch

Mithilfe von CloudWatch Metriken können Sie Ihren Instance-Durchsatz beobachten und herausfinden, ob Ihre Instance-Klasse ausreichend Ressourcen für Ihre Anwendungen bereitstellt. Informationen zu Ihren Instance-Klassenlimits finden Sie in den Spezifikationen für Ihre Instance-Klasse, um Ihre Netzwerkleistung zu ermitteln. Instance-Limits

Wenn Ihre Instance-Nutzung nahe dem Instance-Klassenlimit liegt, kann es sein, dass sich die Leistung verlangsamt. Die CloudWatch Metriken können diese Situation bestätigen, sodass Sie planen können, manuell auf eine größere Instance-Klasse hochzuskalieren.

Kombinieren Sie die folgenden CloudWatch Metrikwerte, um herauszufinden, ob Sie sich dem Instance-Klassenlimit nähern:

- NetworkThroughput— Die Menge des Netzwerkdurchsatzes, der von den Clients für jede Instance im Amazon DocumentDB-Cluster empfangen und übertragen wird. Dieser Durchsatzwert beinhaltet nicht den Netzwerkverkehr zwischen Instances im Cluster und dem Cluster-Speichervolumen.
- StorageNetworkThroughput— Die Menge des Netzwerkdurchsatzes, der von jeder Instance im Amazon DocumentDB-Cluster empfangen und an das Amazon DocumentDB-Cluster-Speichervolumen gesendet wurde.

Fügen Sie den hinzu NetworkThroughput, um den Netzwerkdurchsatz StorageNetworkThroughputzu ermitteln, der von jeder Instance in Ihrem Amazon DocumentDB-Cluster vom Amazon DocumentDB-Cluster empfangen und an das Amazon DocumentDB-Cluster gesendet wurde. Das Instance-Klassenlimit für Ihre Instance sollte größer sein als die Summe dieser beiden kombinierten Metriken.

Sie können die folgenden Metriken verwenden, um zusätzliche Details des Netzwerkverkehrs Ihrer Client-Anwendungen beim Senden und Empfangen zu überprüfen:

- NetworkReceiveThroughput— Die Menge des Netzwerkdurchsatzes, den jede Instance im Amazon DocumentDB-Cluster von Clients erhält. Dieser Durchsatz beinhaltet nicht den Netzwerkverkehr zwischen Instances im Cluster und dem Cluster-Speichervolumen.
- NetworkTransmitThroughput— Die Menge des Netzwerkdurchsatzes, der von jeder Instance im Amazon DocumentDB-Cluster an Clients gesendet wird. Dieser Durchsatz beinhaltet nicht den Netzwerkverkehr zwischen Instances im Cluster und dem Cluster-Speichervolumen.

- StorageNetworkReceiveThroughput— Die Menge des Netzwerkdurchsatzes, den jede Instance im Cluster vom Amazon DocumentDB-Cluster-Speichervolumen erhält.
- StorageNetworkTransmitThroughput— Die Menge des Netzwerkdurchsatzes, der von jeder Instance im Cluster an das Amazon DocumentDB-Cluster-Speichervolume gesendet wird.

Addieren Sie all diese Metriken, um zu bewerten, wie Ihre Netzwerknutzung im Vergleich zum Instance-Klassenlimit abschneidet. Das Instance-Klassenlimit sollte größer sein als die Summe dieser kombinierten Metriken.

Die Netzwerklimits und die CPU-Auslastung einer Instance sind wechselseitig. Wenn der Netzwerkdurchsatz steigt, steigt auch die CPU-Auslastung. Die Überwachung der CPU- und Netzwerknutzung gibt Aufschluss darüber, wie und warum die Ressourcen erschöpft werden.

Sie können Folgendes in Betracht ziehen, um die Netzwerknutzung zu minimieren:

- Verwenden einer größeren Instance-Klasse
- Aufteilen der Schreibanforderungen in Batches, um die Gesamttransaktionen zu reduzieren
- Weiterleiten der schreibgeschützten Workload an eine schreibgeschützte Instance
- Löschen aller ungenutzten Indizes

Optimieren von Abfragen

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

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

Sie können den Befehl `explain` auch verwenden, um zu erfahren, wie ein Abfrageplan für eine bestimmte Abfrage analysiert wird. Mithilfe dieser Informationen können Sie eine Abfrage oder zugrundeliegende Sammlung ändern, um die Abfrageleistung zu verbessern (z. B. Hinzufügen eines Indexes).

TTL- und Zeitreihen-Workloads

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

Wenn der TTL-Monitor Ihre Dokumente löscht, fallen bei jedem Löschen I/O Kosten an, was Ihre Rechnung in die Höhe treibt. Wenn der Durchsatz und die TTL-Löschraten steigen, sollten Sie aufgrund der erhöhten Nutzung mit einer höheren Rechnung rechnen. I/O Wenn Sie jedoch keinen TTL-Index zum Löschen von Dokumenten erstellen, sondern Dokumente anhand der Zeit in Sammlungen unterteilen und diese Sammlungen einfach löschen, wenn sie nicht mehr benötigt werden, fallen Ihnen keine I/O-Kosten an. Dies kann erheblich kostengünstiger sein als die Verwendung eines TTL-Index.

Für Zeitreihen-Workloads können Sie erwägen, statt eines TTL-Index fortlaufende Sammlungen zu erstellen, da rollierende Sammlungen eine bessere und kostengünstigere Methode zum Löschen von Daten sein können und zudem schneller und kostengünstiger sein können als die Verwendung eines TTL-Indexes. I/O intensive. If you have large collections (especially collections over 1TB) or TTL deletion I/O costs are a concern, we recommend that you partition documents into collections based on time, and drop collections when the documents are no longer needed. You can create one collection per day or one per week, depending on your data ingest rate. While requirements will vary depending on your application, a good rule of thumb is to have more smaller collections rather than a few large collections. Dropping these collections does not incur I/O

Migrationen

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

Wir empfehlen außerdem, Ihre Anwendung vor der Migration Ihrer Produktionsdatenbank vollständig auf Amazon DocumentDB zu testen und dabei Funktionalität, Leistung, Betrieb und Kosten zu berücksichtigen.

Arbeiten mit Cluster-Parametergruppen

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

Abfragen in der Aggregationspipeline

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

`batchInsert` und **`batchUpdate`**

Wenn Sie eine hohe Anzahl an gleichzeitigen `batchUpdate` Vorgängen `batchInsert` und/oder Vorgängen ausführen und die Anzahl von `FreeableMemory` (CloudWatch Metric) auf Ihrer primären Instance auf Null sinkt, können Sie entweder die Parallelität der Batch-Insert- oder Aktualisierungs-Workloads reduzieren oder, falls die Parallelität der Arbeitslast nicht reduziert werden kann, die Instance-Größe erhöhen, um die Menge von zu erhöhen. `FreeableMemory`

Funktionale Unterschiede: Amazon DocumentDB und MongoDB

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

Themen

- Funktionale Vorteile von Amazon DocumentDB
- Funktionelle Unterschiede wurden aktualisiert
- Funktionale Unterschiede zu MongoDB

Funktionale Vorteile von Amazon DocumentDB

Implizite Transaktionen

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

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

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

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

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

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

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

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

Funktionelle Unterschiede wurden aktualisiert

Amazon DocumentDB verbessert weiterhin die Kompatibilität mit MongoDB, indem es von den Funktionen, die unsere Kunden uns in Auftrag gegeben haben, abweicht. Dieser Abschnitt enthält die funktionalen Unterschiede, die wir in Amazon DocumentDB entfernt haben, um unseren Kunden Migrationen und die Erstellung von Anwendungen zu erleichtern.

Themen

- Array-Indizierung
- Indizes mit mehreren Schlüsseln
- Nullzeichen in Zeichenketten
- Rollenbasierte Zugriffskontrolle
- $regexIndizierung
- Projektion für verschachtelte Dokumente

Array-Indizierung

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

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

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

Mit folgendem Befehl können Sie die Version Ihrer Indizes abfragen.

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

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

```
[
```

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

Indizes mit mehreren Schlüsseln

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

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

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

Mit folgendem Befehl können Sie die Version Ihrer Indizes abfragen.

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

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

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

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

Nullzeichen in Zeichenketten

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

Rollenbasierte Zugriffskontrolle

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

`$regex`Indizierung

Seit dem 22. Juni 2020 unterstützt Amazon DocumentDB nun die Möglichkeit für `$regex` Betreiber, einen Index zu verwenden.

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

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

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

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

Projektion für verschachtelte Dokumente

Es gibt einen funktionellen Unterschied zwischen Amazon DocumentDB und MongoDB in Version 3.6, der in Amazon DocumentDB 4.0 behoben wurde, aber in Amazon DocumentDB 3.6 weiterhin nicht unterstützt wird. `$project`

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

Beispiel: Wenn die Projektion ist`"a.b.c": 1`, dann funktioniert das Verhalten sowohl in Amazon DocumentDB als auch in MongoDB erwartungsgemäß. Wenn die Projektion jedoch `{a:{b:{c:1}}}` dann ist, wendet Amazon DocumentDB 3.6 die Projektion nur auf an `a` und nicht auf `b` oder`c`. In Amazon DocumentDB 4.0 `{a:{b:{c:1}}}` wird die Projektion auf `ab`, und `c` angewendet.

Funktionale Unterschiede zu MongoDB

Themen

- $vectorSearch-Operator
- OpCountersCommand
- Admin-Datenbanken und Sammlungen
- cursormaxTimeMS
- explain()
- Der Index wird erstellt
- Suche mit leerem Schlüssel im Pfad
- MongoDB APIs, Operationen und Datentypen
- mongodumpund mongorestore Dienstprogramme
- Reihenfolge der Ergebnisse
- Wiederholbare Schreibvorgänge

- Sparse-Index
- $elemMatchVerwendung innerhalb eines Ausdrucks $all
- $ne,,$nin, nornot$exists, und $elemMatch Indizierung
- Dollar ($) und Punkt (.) in Feldnamen
- $lookup
- $naturalund umgekehrte Sortierung

`$vectorSearch`-Operator

Amazon DocumentDB unterstützt nicht `$vectorSearch` als unabhängiger Betreiber. Stattdessen unterstützen wir `vectorSearch` innerhalb des `$search` Betreibers. Weitere Informationen finden Sie unter Vektorsuche für Amazon DocumentDB.

`OpCountersCommand`

Das `OpCountersCommand` Verhalten von Amazon DocumentDB unterscheidet sich wie folgt von MongoDB: `opcounters.command`

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

Admin-Datenbanken und Sammlungen

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

`cursormaxTimeMS`

`cursor.maxTimeMS`Setzt in Amazon DocumentDB den Zähler für jede `getMore` Anfrage zurück. Wenn also ein Wert von 3000 MS angegeben `maxTimeMS` ist, die Abfrage 2800 MS und jede nachfolgende `getMore` Anforderung 300 MS benötigt, wird der Cursor kein Timeout haben. Für den Cursor wird nur ein Timeout ausgeführt, wenn eine einzelne Operation, entweder die Abfrage oder eine einzelne `getMore` Anfrage, länger dauert als angegeben. `maxTimeMS` Außerdem läuft der Sweeper, der die Cursor-Ausführungszeit überprüft, mit einer Granularität von fünf (5) Minuten.

explain()

Amazon DocumentDB emuliert MongoDB 3.6, 4.0 und 5.0 APIs auf einer speziell entwickelten Datenbank-Engine, die ein verteiltes, fehlertolerantes, selbstheilendes Speichersystem verwendet. Daher `explain()` können sich die Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterscheiden. Kunden, die die Kontrolle über ihren Abfrageplan wünschen, können den `$hint`-Operator verwenden, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Der Index wird erstellt

Amazon DocumentDB lässt zu, dass jeweils nur ein Index für eine Sammlung erstellt wird. Entweder im Vordergrund oder im Hintergrund. Wenn Vorgänge wie `createIndex()` oder `dropIndex()` für dieselbe Sammlung auftreten, wenn gerade ein Index erstellt wird,, schlägt der neu versuchte Vorgang fehl.

Standardmäßig werden Indexerstellungen in Amazon DocumentDB und MongoDB Version 4.0 im Hintergrund ausgeführt. MongoDB Version 4.2 und höher ignoriert die Option zum Erstellen des Hintergrundindexes, wenn sie für createIndexes oder seine Shell-Helfer und angegeben wurde. `createIndex()` `createIndexes()`

Bei einem Time to Live (TTL) -Index laufen Dokumente ab, nachdem die Indexerstellung abgeschlossen ist.

Suche mit leerem Schlüssel im Pfad

Wenn Sie mit einem Schlüssel suchen, der eine leere Zeichenfolge als Teil des Pfads enthält (z. B.`x..b`)`x.`, und das Objekt einen leeren Zeichenkettenschlüsselpfad (z. B.`{"x" : [ { "" : 10 }, { "b" : 20 } ]}`) innerhalb eines Arrays hat, gibt Amazon DocumentDB andere Ergebnisse zurück, als wenn Sie dieselbe Suche in MongoDB ausführen würden.

In MongoDB funktioniert die Suche nach einem leeren Schlüsselpfad innerhalb eines Arrays wie erwartet, wenn sich der leere Zeichenkettenschlüssel nicht am Ende der Pfadsuche befindet. Wenn sich der leere Zeichenkettenschlüssel jedoch am Ende der Pfadsuche befindet, schaut er nicht in das Array hinein.

In Amazon DocumentDB wird jedoch nur das erste Element innerhalb des Arrays gelesen, da eine leere Zeichenfolge in eine leere Zeichenfolge `getArrayIndexFromKeyString` konvertiert wird`0`,

sodass die Suche nach Zeichenkettenschlüsseln wie die Suche nach einem Array-Index behandelt wird.

MongoDB APIs, Operationen und Datentypen

Amazon DocumentDB ist mit MongoDB 3.6, 4.0 und 5.0 kompatibel. APIs Eine up-to-date Liste der unterstützten Funktionen finden Sie unter. Unterstützte MongoDB APIs, Operationen und Datentypen in Amazon DocumentDB

`mongodump`und **`mongorestore`** Dienstprogramme

Amazon DocumentDB unterstützt keine Admin-Datenbank und speichert die Admin-Datenbank daher nicht und stellt sie auch nicht wieder her, wenn Sie die Dienstprogramme `mongodump` oder `mongorestore` verwenden. Wenn Sie mit Amazon DocumentDB eine neue Datenbank erstellen`mongorestore`, müssen Sie zusätzlich zum Wiederherstellungsvorgang auch die Benutzerrollen neu erstellen.

> **Note**
>
> Wir empfehlen MongoDB Database Tools bis einschließlich Version 100.6.1 für Amazon DocumentDB. Sie können hier auf die Downloads der MongoDB Database Tools zugreifen.

Reihenfolge der Ergebnisse

Amazon DocumentDB garantiert nicht die implizite Sortierreihenfolge von Ergebnismengen. Um die Reihenfolge einer Ergebnismenge zu gewährleisten, geben Sie mit `sort()` explizit eine Sortierreihenfolge an.

Das folgende Beispiel sortiert die Artikel in der Inventur in absteigender Reihenfolge basierend auf dem Bestandsfeld.

```
db.inventory.find().sort({ stock: -1 })
```

Bei Verwendung der `$sort` Aggregationsphase wird die Sortierreihenfolge nicht beibehalten, es sei denn, die `$sort` Phase ist die letzte Phase in der Aggregationspipeline. Wenn die `$sort` Aggregationsphase in Kombination mit der Aggregationsphase verwendet wird, wird die `$group` Aggregationsphase nur auf die `$sort` Akkumulatoren und angewendet. `$first` `$last` In Amazon

DocumentDB 4.0 wurde Unterstützung für `$push` die Beibehaltung der Sortierreihenfolge aus der vorherigen `$sort` Phase hinzugefügt.

Wiederholbare Schreibvorgänge

Ab MongoDB 4.2-kompatiblen Treibern sind wiederholbare Schreibvorgänge standardmäßig aktiviert. Amazon DocumentDB unterstützt derzeit jedoch keine wiederholbaren Schreibvorgänge. Der funktionale Unterschied wird sich in einer Fehlermeldung ähnlich der folgenden zeigen.

```
{"ok":0,"errmsg":"Unrecognized field: 'txnNumber'","code":9,"name":"MongoError"}
```

Wiederholbare Schreibvorgänge können über die Verbindungszeichenfolge (z. B.`MongoClient("mongodb://my.mongodb.cluster/db?retryWrites=false")`) oder das Schlüsselwortargument des MongoClient Konstruktors (z. B.) deaktiviert werden. `MongoClient("mongodb://my.mongodb.cluster/db", retryWrites=False)`

Im Folgenden finden Sie ein Python-Beispiel, das wiederholbare Schreibvorgänge in der Verbindungszeichenfolge deaktiviert.

```
client =
 pymongo.MongoClient('mongodb://
<username>:<password>@docdb-2019-03-17-16-49-12.cluster-ccuszbx3pn5e.us-
east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0',w='majority',j=True,retryWrites=False)
```

Sparse-Index

Um einen Sparse-Index zu verwenden, den Sie in einer Abfrage erstellt haben, müssen Sie die `$exists`-Bedingungen für die Felder verwenden, die der Index abdecken. Wenn Sie ihn weglassen`$exists`, verwendet Amazon DocumentDB den Sparse-Index nicht.

Im Folgenden wird ein Beispiel gezeigt.

```
db.inventory.count({ "stock": { $exists: true }})
```

Für spärliche Indizes mit mehreren Schlüsseln unterstützt Amazon DocumentDB keine eindeutige Schlüsseleinschränkung, wenn die Suche nach einem Dokument zu einer Reihe von Werten führt und nur eine Teilmenge der indizierten Felder fehlt. Beispielsweise wird `createIndex({"a.b" : 1 }, { unique : true, sparse :true })` nicht unterstützt, wenn `"a" : [ { "b" : 2 }, { "c" : 1 } ]` eingegeben wird, da `"a.c"` im Index gespeichert ist.

$elemMatch Verwendung innerhalb eines Ausdrucks **$all**

Amazon DocumentDB unterstützt derzeit nicht die Verwendung des `$elemMatch` Operators innerhalb eines `$all` Ausdrucks. Sie können dies umgehen, indem Sie den Operator `$and` wie folgt mit `$elemMatch` verwenden.

Ursprüngliche Operation:

```
db.col.find({
  qty: {
    $all: [
      { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } },
      { "$elemMatch": { num: 40, size: "XL" } }
    ]
  }
})
```

Aktualisierte Operation:

```
db.col.find({
  $and: [
    { qty: { "$elemMatch": { part: "xyz", qty: { $lt: 11 } } } },
    { qty: { "$elemMatch": { qty: 40, size: "XL" } } }
  ]
})
```

$ne,,**$nin**, **nornot$exists**, und **$elemMatch** Indizierung

Amazon DocumentDB unterstützt derzeit nicht die Möglichkeit, Indizes mit den Operatoren`$ne`,,`$nin`, `$nor` `$not$exists`, und `$distinct` zu verwenden. Aus diesem Grund führt die Verwendung dieser Operatoren zu Sammelscans. Wenn Sie vor der Verwendung eines dieser Operatoren einen Filter oder einen Abgleich durchführen, wird die Datenmenge reduziert, die gescannt werden muss, und kann somit die Leistung verbessern.

Amazon DocumentDB hat Unterstützung für Indexscans mit dem `$elemMatch` Operator in Amazon DocumentDB 5.0 und elastischen Clustern hinzugefügt. Indexscans werden unterstützt, wenn der reine Abfragefilter eine `$elemMatch` Filterebene hat, aber nicht unterstützt, wenn eine verschachtelte `$elemMatch` Abfrage enthalten ist.

`$elemMatch`Abfrageform, die Indexscans in Amazon DocumentDB 5.0 unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

`$elemMatch`Abfrageform, die in Amazon DocumentDB 5.0 keine Indexscans unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e": "abc"} }} } })
```

Dollar ($) und Punkt (.) in Feldnamen

Amazon DocumentDB unterstützt nicht die Abfrage von Feldern mit dem Präfix Dollar ($) in $in, $nin und $all in verschachtelten Objekten. Die folgende Abfrage ist beispielsweise in Amazon DocumentDB nicht gültig:

```
coll.find({"field": {"$all": [{ "$a": 1 }]}})
```

`$lookup`

Amazon DocumentDB unterstützt Gleichheitsabgleiche (z. B. Left Outer Join) und unterstützt auch unkorrelierte Unterabfragen, aber keine korrelierten Unterabfragen.

Verwenden eines Indexes mit `$lookup`

Sie können jetzt einen Index mit dem `$lookup` Stage-Operator verwenden. Je nach Anwendungsfall gibt es mehrere Indexierungsalgorithmen, mit denen Sie die Leistung optimieren können. In diesem Abschnitt werden die verschiedenen Indexierungsalgorithmen für Sie erläutert `$lookup` und Sie bei der Auswahl des für Ihren Workload am besten geeigneten Algorithmus unterstützt.

Standardmäßig verwendet Amazon DocumentDB den Hash-Algorithmus, wenn `allowDiskUse:false` es verwendet wird, und Sort Merge, wenn `allowDiskUse:true` es verwendet wird.

> **Note**
>
> Die `allowDiskUse` Option wird derzeit für den `find` Befehl nicht unterstützt. Die Option wird nur als Teil der Aggregation unterstützt. Wir empfehlen, das Aggregationsframework mit `allowDiskUse:true` zu verwenden, um große Abfragen zu verarbeiten, die Speicherlimits überschreiten könnten.

In einigen Anwendungsfällen kann es wünschenswert sein, den Abfrageoptimierer zu zwingen, einen anderen Algorithmus zu verwenden. Im Folgenden sind die verschiedenen Indizierungsalgorithmen aufgeführt, die der `$lookup` Aggregationsoperator verwenden kann:

- Verschachtelte Schleife: Ein Plan mit verschachtelter Schleife ist in der Regel für eine Arbeitslast von Vorteil, wenn die ausländische Sammlung weniger als 1 GB groß ist und das Feld in der ausländischen Sammlung über einen Index verfügt. Wenn der Nested-Loop-Algorithmus verwendet wird, zeigt der Explain-Plan die Phase als an. `NESTED_LOOP_LOOKUP`
- Zusammenführung sortieren: Ein Plan zur Zusammenführung von Sortierungen ist in der Regel für eine Arbeitslast von Vorteil, wenn die ausländische Sammlung keinen Index für das bei der Suche verwendete Feld hat und der Arbeitsdatensatz nicht in den Arbeitsspeicher passt. Wenn der Algorithmus zur Sortierung und Zusammenführung verwendet wird, zeigt der Erläuterungsplan die Phase als an`SORT_LOOKUP`.
- Hash: Ein Hash-Plan ist in der Regel für eine Arbeitslast von Vorteil, wenn die ausländische Sammlung weniger als 1 GB groß ist und der Arbeitsdatensatz in den Arbeitsspeicher passt. Wenn der Hash-Algorithmus verwendet wird, zeigt der Explain-Plan die Phase als an`HASH_LOOKUP`.

Sie können den Indizierungsalgorithmus, der für den `$lookup` Operator verwendet wird, `explain` anhand der Abfrage identifizieren. Im Folgenden finden Sie ein Beispiel:

```
db.localCollection.explain().aggregate(
   [
      {
         $lookup:
            {
               from: "foreignCollection",
               localField: "a",
               foreignField: "b",
               as: "joined"
            }
      }
   ]
)

output
{
    "queryPlanner" : {
        "plannerVersion" : 1,
```

```
        "namespace" : "test.localCollection",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "SORT_AGGREGATE",
                "inputStage" : {
                    "stage" : "SORT",
                    "inputStage" : {
                        "stage" : "NESTED_LOOP_LOOKUP",
                        "inputStages" : [
                            {
                                "stage" : "COLLSCAN"
                            },
                            {
                                "stage" : "FETCH",
                                "inputStage" : {
                                    "stage" : "COLLSCAN"
                                }
                            }
                        ]
                    }
                }
            }
        }
    },
    "serverInfo" : {
        "host" : "devbox-test",
        "port" : 27317,
        "version" : "3.6.0"
    },
    "ok" : 1
}
```

Als Alternative zur Verwendung der `explain()` Methode können Sie den Profiler verwenden, um den Algorithmus zu überprüfen, der bei Ihrer Verwendung des `$lookup` Operators verwendet wird. Weitere Informationen zum Profiler finden Sie unter. Profilierung von Amazon DocumentDB DocumentDB-Vorgängen

Verwenden eines **`planHint`**

Wenn Sie den Abfrageoptimierer zwingen möchten, einen anderen Indexierungsalgorithmus mit zu verwenden`$lookup`, können Sie einen verwenden. `planHint` Verwenden Sie dazu den Kommentar

in den Optionen der Aggregationsphase, um einen anderen Plan zu erzwingen. Im Folgenden finden Sie ein Beispiel für die Syntax des Kommentars:

```
comment : {
    comment :  "<string>",
    lookupStage : { planHint : "SORT" | "HASH" | "NESTED_LOOP" }
}
```

Im Folgenden finden Sie ein Beispiel für die Verwendung von`planHint`, um den Abfrageoptimierer zur Verwendung des HASH Indexierungsalgorithmus zu zwingen:

```
db.foo.aggregate(
   [
      {
         $lookup:
            {
               from: "foo",
               localField: "_id",
               foreignField: "_id",
               as: "joined"
            },
      }
   ]
),
{
   comment : "{ \"lookupStage\" : { \"planHint\": \"HASH\" }}"
```

Um zu testen, welcher Algorithmus für Ihre Arbeitslast am besten geeignet ist, können Sie den `executionStats` Parameter der `explain` Methode verwenden, um die Ausführungszeit der `$lookup` Phase zu messen und gleichzeitig den Indizierungsalgorithmus zu ändern (d. h.HASH//SORT). NESTED_LOOP

Das folgende Beispiel zeigt, wie Sie `executionStats` die Ausführungszeit der `$lookup` Phase mithilfe des SORT Algorithmus messen können.

```
db.foo.explain("executionStats").aggregate(
   [
      {
         $lookup:
```

```
            {
               from: "foo",
               localField: "_id",
               foreignField: "_id",
               as: "joined"
            },
      }
   ]
),
{
   comment : "{ \"lookupStage\" : { \"planHint\": \"SORT\" }}"
```

$naturalund umgekehrte Sortierung

Amazon DocumentDB unterstützt nur `$natural` Forward-Collection-Scans. Rückwärtige Sammelscans (`{$natural: -1}`) führen zu einem`MongoServerError`.

Unterstützte MongoDB APIs, Operationen und Datentypen in Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein schneller, skalierbarer, hochverfügbarer und vollständig verwalteter Dokumentendatenbankservice, der MongoDB-Workloads unterstützt. Amazon DocumentDB ist mit MongoDB 3.6, 4.0 und 5.0 kompatibel. APIs In diesem Abschnitt werden die unterstützten Funktionalitäten aufgeführt. Unterstützung bei der Verwendung von MongoDB APIs und Treibern finden Sie in den MongoDB-Community-Foren. Wenn Sie Unterstützung bei der Nutzung des Amazon DocumentDB-Services benötigen, wenden Sie sich bitte an das entsprechende AWS Support-Team. Informationen zu den Funktionsunterschieden zwischen Amazon DocumentDB und MongoDB finden Sie unter. Funktionale Unterschiede: Amazon DocumentDB und MongoDB

MongoDB-Befehle und -Operatoren, die nur intern oder nicht auf einen verwalteten Service anwendbar sind, werden nicht unterstützt und sind nicht in der Liste der unterstützten Funktionalität enthalten.

Seit der Markteinführung haben wir mehr als 50 zusätzliche Funktionen hinzugefügt und werden uns weiter an den Anforderungen unserer Kunden ausrichten, um ihnen die benötigten Funktionen bereitzustellen. Informationen zu den neuesten Produkteinführungen finden Sie unter Ankündigungen von Amazon DocumentDB.

Wenn es eine Funktion gibt, die nicht unterstützt wird und die Sie von uns erstellen lassen möchten, teilen Sie uns dies mit, indem Sie eine E-Mail mit Ihrer accountID, den angeforderten Funktionen und dem Anwendungsfall an das Amazon DocumentDB-Serviceteam senden.

Themen

- Datenbankbefehle
- Abfrage- und Projektionsoperatoren
- Operatoren aktualisieren
- Geodaten
- Cursor-Methoden
- Operatoren für Aggregationspipelines
- Datentypen

- Indizes und Indexeigenschaften

Datenbankbefehle

Themen

- Administrative Befehle
- Aggregation
- Authentifizierung
- Diagnosebefehle
- Abfrage- und Schreiboperationen
- Befehle zur Rollenverwaltung
- Befehle für Sitzungen
- Benutzerverwaltung
- Sharding-Befehle

Administrative Befehle

Befehl	3.6	4,0	5.0	Elastischer Cluster
Gedeckelte Sammlungen	Nein	Nein	Nein	Nein
cloneCollectionAsGekappt	Nein	Nein	Nein	Nein
collMod	Teilweise	Teilweise	Teilweise	Teilweise
collMod: expireAfterSeconds	Ja	Ja	Ja	Ja
convertToCapped	Nein	Nein	Nein	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
copydb	Nein	Nein	Nein	Nein
create	Ja	Ja	Ja	Ja
createView	Nein	Nein	Nein	Nein
createIndexes	Ja	Ja	Ja	Ja
currentOp	Ja	Ja	Ja	Ja
fallen lassen	Ja	Ja	Ja	Ja
dropDatabase	Ja	Ja	Ja	Ja
dropIndexes	Ja	Ja	Ja	Ja
filemd5	Nein	Nein	Nein	Nein
getAuditConfig	Nein	Ja	Ja	Nein
killCursors	Ja	Ja	Ja	Ja
killOp	Ja	Ja	Ja	Ja
listCollections auflisten*	Ja	Ja	Ja	Ja
listDatabases	Ja	Ja	Ja	Ja
listIndexes	Ja	Ja	Ja	Ja
reIndex	Nein	Nein	Ja	Nein
renameCol lection	Ja	Ja	Ja	Nein
setAuditConfig	Nein	Ja	Ja	Nein

* Der `type` Schlüssel in der Filteroption wird nicht unterstützt.

Aggregation

Befehl	3.6	4,0	5.0	Elastischer Cluster
aggregate	Ja	Ja	Ja	Ja
count	Ja	Ja	Ja	Ja
distinct	Ja	Ja	Ja	Ja
mapReduce	Nein	Nein	Nein	Nein

Authentifizierung

Befehl	3.6	4,0	5.0	Elastischer Cluster
authenticate	Ja	Ja	Ja	Ja
logout	Ja	Ja	Ja	Ja

Diagnosebefehle

Befehl	3.6	4,0	5.0	Elastischer Cluster
buildInfo	Ja	Ja	Ja	Ja
collStats	Ja	Ja	Ja	Ja
connPoolStats	Nein	Nein	Nein	Nein
connectionStatus	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
dataSize	Ja	Ja	Ja	Ja
dbHash	Nein	Nein	Nein	Nein
dbStats	Ja	Ja	Ja	Ja
explain	Ja	Ja	Ja	Ja
Erklären: executionStats	Ja	Ja	Ja	Ja
features	Nein	Nein	Nein	Nein
hostInfo	Ja	Ja	Ja	Ja
listCommands	Ja	Ja	Ja	Ja
Profiler	Ja	Ja	Ja	Nein
serverStatus	Ja	Ja	Ja	Ja
top	Ja	Ja	Ja	Ja

Abfrage- und Schreiboperationen

Befehl	3.6	4,0	5.0	Elastischer Cluster
Change streams	Ja	Ja	Ja	Nein
delete	Ja	Ja	Ja	Ja
find	Ja	Ja	Ja	Ja
findAndModify	Ja	Ja	Ja	Ja
getLastError	Nein	Nein	Nein	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
getMore	Ja	Ja	Ja	Ja
getPrevError	Nein	Nein	Nein	Nein
GridFS	Ja	Ja	Ja	Nein
insert	Ja	Ja	Ja	Ja
parallelCollectionScan	Nein	Nein	Nein	Nein
resetError	Nein	Nein	Nein	Nein
update	Ja	Ja	Ja	Ja
ReplaceOne	Ja	Ja	Ja	Ja

Befehle zur Rollenverwaltung

E	3	4	5	E e C
c e	Ja	Ja	Ja	Nein
c l t	Ja	Ja	Ja	Nein a
c	Ja	Ja	Ja	Nein
g s	Ja	Ja	Ja	Nein

E	3	4	5	E e C
r e l	Ja	Ja	Ja	Nein l
r v c	Ja	Ja	Ja	Nein
r	Ja	Ja	Ja	Nein
u e	Ja	Ja	Ja	Nein

Befehle für Sitzungen

Befehl	3.6	4,0	5.0	Elastischer Cluster
Transaktion abbrechen	Nein	Ja	Ja	Nein
commitTransaction	Nein	Ja	Ja	Nein
Sitzungen beenden	Nein	Nein	Nein	Nein
killAllSessions	Nein	Ja	Ja	Nein
killAllSessionsByPattern	Nein	Nein	Nein	Nein
Sitzungen beenden	Nein	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
Sitzungen aktualisieren	Nein	Nein	Nein	Nein
Sitzung starten	Nein	Ja	Ja	Nein

Benutzerverwaltung

Befehl	3.6	4,0	5.0	Elastischer Cluster
createUser	Ja	Ja	Ja	Ja
dropAllUsersFromDatabase	Ja	Ja	Ja	Ja
dropUser	Ja	Ja	Ja	Ja
grantRolesToBenutzer	Ja	Ja	Ja	Ja
revokeRolesFromBenutzer	Ja	Ja	Ja	Ja
updateUser	Ja	Ja	Ja	Ja
Benutzerinformationen	Ja	Ja	Ja	Ja

Sharding-Befehle

Befehl	Elastischer Cluster
abortReshardCollection	Nein

Befehl	Elastischer Cluster
Fügt Hard hinzu	Nein
addShardToZone	Nein
balancerCollectionStatus	Nein
Balancer starten	Nein
Balancer-Status	Nein
Balancer Stopp	Nein
checkShardingIndex	Nein
clearJumboFlag	Nein
cleanupOrphaned	Nein
cleanupReshardCollection	Nein
commitReshardCollection	Nein
Sharding aktivieren	Ja
flushRouterConfig	Nein
getShardMap	Nein
getShardVersion	Nein
isdbgrid	Nein
Listet Scherben auf	Nein
Median-Key	Nein
Chunk verschieben	Nein
Primär verschieben	Nein

Befehl	Elastischer Cluster
Chunks zusammenführen	Nein
refineCollectionShardSchlüssel	Nein
Entfernt Hard	Nein
removeShardFromZone	Nein
Sammlung Reshard	Nein
setAllowMigrations	Nein
setShardVersion	Nein
Sammlung Shard	Ja
Status „Sharding“	Nein
split	Nein
Vektor teilen	Nein
Harding rückgängig machen	Nein
updateZoneKeyReichweite	Nein

Abfrage- und Projektionsoperatoren

Themen

- Array-Operatoren
- Bitweise Operatoren
- Operator kommentieren
- Vergleichsoperatoren
- Elementoperatoren
- Operatoren für Bewertungsabfragen
- Logische Operatoren

- Projektionsoperatoren

Array-Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$all	Ja	Ja	Ja	Ja
$elemMatch	Ja	Ja	Ja	Ja
$size	Ja	Ja	Ja	Ja

Bitweise Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$bitsAllSet	Ja	Ja	Ja	Ja
$bitsAnySet	Ja	Ja	Ja	Ja
$bitsAllClear	Ja	Ja	Ja	Ja
$bitsAnyClear	Ja	Ja	Ja	Ja

Operator kommentieren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$comment	Ja	Ja	Ja	Ja

Vergleichsoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$eq	Ja	Ja	Ja	Ja
$gt	Ja	Ja	Ja	Ja
$gte	Ja	Ja	Ja	Ja
$in	Ja	Ja	Ja	Ja
$lt	Ja	Ja	Ja	Ja
$lte	Ja	Ja	Ja	Ja
$ne	Ja	Ja	Ja	Ja
$nin	Ja	Ja	Ja	Ja

Elementoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$exists	Ja	Ja	Ja	Ja
$type	Ja	Ja	Ja	Ja

Operatoren für Bewertungsabfragen

Befehl	3.6	4,0	5.0	Elastischer Cluster
$expr	Nein	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
$jsonSchema	Nein	Ja	Ja	Nein
$mod	Ja	Ja	Ja	Ja
$regex	Ja	Ja	Ja	Ja
$text	Nein	Nein	Ja	Nein
$where	Nein	Nein	Nein	Nein

Logische Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$and	Ja	Ja	Ja	Ja
$nor	Ja	Ja	Ja	Ja
$not	Ja	Ja	Ja	Ja
$or	Ja	Ja	Ja	Ja

Projektionsoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$	Ja	Ja	Ja	Ja
$elemMatch	Ja	Ja	Ja	Ja
$meta	Nein	Nein	Ja	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
$slice	Ja	Ja	Ja	Ja

Operatoren aktualisieren

Themen

- Array-Operatoren
- Bitweise Operatoren
- Feldoperatoren
- Modifikatoren aktualisieren

Array-Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$	Ja	Ja	Ja	Ja
$[]	Ja	Ja	Ja	Ja
$[<identifier>]	Ja	Ja	Ja	Ja
$addToSet	Ja	Ja	Ja	Ja
$pop	Ja	Ja	Ja	Ja
$pullAll	Ja	Ja	Ja	Ja
$pull	Ja	Ja	Ja	Ja
$push	Ja	Ja	Ja	Ja

Bitweise Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$bit	Ja	Ja	Ja	Ja

Feldoperatoren

Operator	3.6	4,0	5.0	Elastischer Cluster
$currentDate	Ja	Ja	Ja	Ja
$inc	Ja	Ja	Ja	Ja
$max	Ja	Ja	Ja	Ja
$min	Ja	Ja	Ja	Ja
$mul	Ja	Ja	Ja	Ja
$rename	Ja	Ja	Ja	Ja
$set	Ja	Ja	Ja	Ja
$setOnInsert	Ja	Ja	Ja	Ja
$unset	Ja	Ja	Ja	Ja

Modifikatoren aktualisieren

Operator	3.6	4,0	5.0	Elastischer Cluster
$each	Ja	Ja	Ja	Ja

Operator	3.6	4,0	5.0	Elastischer Cluster
$position	Ja	Ja	Ja	Ja
$slice	Ja	Ja	Ja	Ja
$sort	Ja	Ja	Ja	Ja

Geodaten

Geometrie-Spezifizierer

Abfrageauswahl	3.6	4,0	5.0	Elastischer Cluster
$box	Nein	Nein	Nein	Nein
$center	Nein	Nein	Nein	Nein
$centerSphere	Nein	Nein	Nein	Nein
$geometry	Ja	Ja	Ja	Ja
$maxDistance	Ja	Ja	Ja	Ja
$minDistance	Ja	Ja	Ja	Ja
$nearSphere	Ja	Ja	Ja	Ja
$polygon	Nein	Nein	Nein	Nein
$uniqueDocs	Nein	Nein	Nein	Nein

Selektoren abfragen

Befehl	3.6	4,0	5.0	Elastischer Cluster
$geoIntersects	Ja	Ja	Ja	Ja
$geoWithin	Ja	Ja	Ja	Ja
$near	Nein	Nein	Nein	Nein
$nearSphere	Ja	Ja	Ja	Ja
$polygon	Nein	Nein	Nein	Nein
$uniqueDocs	Nein	Nein	Nein	Nein

Cursor-Methoden

Befehl	3.6	4,0	5.0	Elastischer Cluster
cursor.batchSize()	Ja	Ja	Ja	Ja
cursor.close()	Ja	Ja	Ja	Ja
cursor.collation()	Nein	Nein	Nein	Nein
cursor.comment()	Ja	Ja	Ja	Ja
cursor.count()	Ja	Ja	Ja	Ja
cursor.explain()	Ja	Ja	Ja	Nein
cursor.forEach()	Ja	Ja	Ja	Ja
cursor.hasNext()	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
cursor.hint()	Ja	Ja	Ja	Ja*
cursor.isClosed()	Ja	Ja	Ja	Ja
cursor.is Exhausted()	Ja	Ja	Ja	Nein
cursor.itcount()	Ja	Ja	Ja	Nein
cursor.limit()	Ja	Ja	Ja	Nein
cursor.map()	Ja	Ja	Ja	Nein
cursor.max()	Nein	Nein	Nein	Nein
cursor.ma xScan()	Ja	Ja	Ja	Nein
cursor.ma xTimeMS()	Ja	Ja	Ja	Nein
cursor.min()	Nein	Nein	Nein	Nein
cursor.next()	Ja	Ja	Ja	Ja
Cursor. noCursorT imeout()	Nein	Nein	Nein	Nein
Cursor. objsLeftI nBatch ()	Ja	Ja	Ja	Nein
cursor.pretty()	Ja	Ja	Ja	Nein
cursor.re adConcern()	Ja	Ja	Ja	Nein
cursor.readPref()	Ja	Ja	Ja	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
cursor.returnKey()	Nein	Nein	Nein	Nein
Cursor.showRecordId()	Nein	Nein	Nein	Nein
cursor.size()	Ja	Ja	Ja	Nein
cursor.skip()	Ja	Ja	Ja	Nein
cursor.sort()	Ja	Ja	Ja	Nein
cursor.tailable()	Nein	Nein	Nein	Nein
cursor.toArray()	Ja	Ja	Ja	Nein

* Index `hint` wird mit Indexausdrücken unterstützt. Beispiel, `db.foo.find().hint({x:1})`.

Operatoren für Aggregationspipelines

Themen

- Akkumulatorausdrücke
- Arithmetische Operatoren
- Array-Operatoren
- Boolesche Operatoren
- Vergleichsoperatoren
- Operatoren für bedingte Ausdrücke
- Datentyp-Operator
- Operator zur Datengröße
- Datumsoperatoren
- Wörtlicher Operator
- Operator zusammenführen

- Natürlicher Operator
- Satzoperatoren
- Bühnenbetreiber
- Zeichenfolgenoperatoren
- Systemvariablen
- Operator für die Textsuche
- Geben Sie Konvertierungsoperatoren ein
- Operatoren für Variablen
- Verschiedene Betreiber

Akkumulatorausdrücke

Expression	3.6	4,0	5.0	Elastischer Cluster
$Akkumulator	-	-	Nein	Nein
$addToSet	Ja	Ja	Ja	Ja
$avg	Ja	Ja	Ja	Ja
$count	-	-	Nein	Nein
$KovarianzPop	Nein	Nein	Nein	Nein
$KovarianzAMP	Nein	Nein	Nein	Nein
$ DenseRank	Nein	Nein	Nein	Nein
$ Ableitung	Nein	Nein	Nein	Nein
$ Nummer des Dokuments	Nein	Nein	Nein	Nein
$expMovingAvg	Nein	Nein	Nein	Nein
$first	Ja	Ja	Ja	Ja

Expression	3.6	4,0	5.0	Elastischer Cluster
$ integral	Nein	Nein	Nein	Nein
$last	Ja	Ja	Ja	Ja
$max	Ja	Ja	Ja	Ja
$min	Ja	Ja	Ja	Ja
$push	Ja	Ja	Ja	Ja
$ Rang	Nein	Nein	Nein	Nein
$ Schicht	Nein	Nein	Nein	Nein
$stdDevPop	Nein	Nein	Nein	Nein
$stdDevSamp	Nein	Nein	Nein	Nein
$sum	Ja	Ja	Ja	Ja

Arithmetische Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$abs	Ja	Ja	Ja	Ja
$add	Ja	Ja	Ja	Ja
$ceil	Nein	Ja	Ja	Ja
$divide	Ja	Ja	Ja	Ja
$exp	Nein	Ja	Ja	Ja
$floor	Nein	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
$ln	Nein	Ja	Ja	Ja
$log	Nein	Ja	Ja	Ja
$log10	Nein	Ja	Ja	Ja
$mod	Ja	Ja	Ja	Ja
$multiply	Ja	Ja	Ja	Ja
$pow	Nein	Nein	Nein	Nein
$rund	-	-	Nein	Nein
$sqrt	Nein	Ja	Ja	Ja
$subtract	Ja	Ja	Ja	Ja
$trunc	Nein	Nein	Nein	Nein

Array-Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$arrayElemAt	Ja	Ja	Ja	Ja
$arrayToObject	Ja	Ja	Ja	Ja
$concatArrays	Ja	Ja	Ja	Ja
$filter	Ja	Ja	Ja	Ja
$first	-	-	Nein	Nein
$in	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
$indexOfArray	Ja	Ja	Ja	Ja
$isArray	Ja	Ja	Ja	Ja
$last	-	-	Nein	Nein
$objectToArray	Ja	Ja	Ja	Ja
$range	Ja	Ja	Ja	Ja
$reverseArray	Ja	Ja	Ja	Ja
$reduce	Ja	Ja	Ja	Ja
$size	Ja	Ja	Ja	Ja
$slice	Ja	Ja	Ja	Ja
$zip	Ja	Ja	Ja	Ja

Boolesche Operatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$and	Ja	Ja	Ja	Ja
$not	Ja	Ja	Ja	Ja
$or	Ja	Ja	Ja	Ja

Vergleichsoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$cmp	Ja	Ja	Ja	Ja
$eq	Ja	Ja	Ja	Ja
$gt	Ja	Ja	Ja	Ja
$gte	Ja	Ja	Ja	Ja
$lt	Ja	Ja	Ja	Ja
$lte	Ja	Ja	Ja	Ja
$ne	Ja	Ja	Ja	Ja

Operatoren für bedingte Ausdrücke

Befehl	3.6	4,0	5.0	Elastischer Cluster
$cond	Ja	Ja	Ja	Ja
$ifNull	Ja	Ja	Ja	Ja
$switch	Nein	Ja	Ja	Nein

Datentyp-Operator

Befehl	3.6	4,0	5.0	Elastischer Cluster
$type	Ja	Ja	Ja	Ja

Operator zur Datengröße

Befehl	3.6	4,0	5.0	Elastischer Cluster
$BinarySize	-	-	Nein	Nein
$ bsonGröße	-	-	Nein	Nein

Datumsoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$DateAdd	Nein	Nein	Ja	Ja
$dateDiff	-	-	Nein	Nein
$dateFromParts	Nein	Nein	Nein	Nein
$dateFromString	Ja	Ja	Ja	Ja
$DateSubtrahieren	Nein	Nein	Ja	Ja
$dateToParts	Nein	Nein	Nein	Nein
$dateToString	Ja	Ja	Ja	Ja
$DateTrunc	-	-	Nein	Nein
$dayOfMonth	Ja	Ja	Ja	Ja
$dayOfWeek	Ja	Ja	Ja	Ja
$dayOfYear	Ja	Ja	Ja	Ja
$hour	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
$ Woche isoDayOf	Ja	Ja	Ja	Ja
$isoWeek	Ja	Ja	Ja	Ja
$isoWeekYear	Ja	Ja	Ja	Ja
$millisecond	Ja	Ja	Ja	Ja
$minute	Ja	Ja	Ja	Ja
$month	Ja	Ja	Ja	Ja
$second	Ja	Ja	Ja	Ja
$week	Ja	Ja	Ja	Ja
$year	Ja	Ja	Ja	Ja

Wörtlicher Operator

Befehl	3.6	4,0	5.0	Elastischer Cluster
$literal	Ja	Ja	Ja	Ja

Operator zusammenführen

Befehl	3.6	4,0	5.0	Elastischer Cluster
$mergeObjects	Ja	Ja	Ja	Ja

Natürlicher Operator

Befehl	3.6	4,0	5.0	Elastischer Cluster
$ natürlich	Ja	Ja	Ja	Ja

Satzoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$allElementsTrue	Nein	Ja	Ja	Ja
$anyElementTrue	Nein	Ja	Ja	Ja
$setDifference	Nein	Ja	Ja	Ja
$setEquals	Ja	Ja	Ja	Ja
$setIntersection	Ja	Ja	Ja	Ja
$setIsSubset	Ja	Ja	Ja	Ja
$setUnion	Ja	Ja	Ja	Ja
$setWindowFields	Nein	Nein	Nein	Nein

Bühnenbetreiber

Befehl	3.6	4,0	5.0	Elastischer Cluster
$addFields	Ja	Ja	Ja	Ja
$bucket	Nein	Nein	Nein	Nein
$bucketAuto	Nein	Nein	Nein	Nein
$collStats	Nein	Nein	Nein	Nein
$count	Ja	Ja	Ja	Ja
$currentOp	Ja	Ja	Ja	Ja
$facet	Nein	Nein	Nein	Nein
$geoNear	Ja	Ja	Ja	Ja
$graphLookup	Nein	Nein	Nein	Nein
$group	Ja	Ja	Ja	Ja
$indexStats	Ja	Ja	Ja	Ja
$limit	Ja	Ja	Ja	Ja
$listLocalSessions	Nein	Nein	Nein	Nein
$listSessions	Nein	Nein	Nein	Nein
$lookup	Ja	Ja	Ja	Ja
$match	Ja	Ja	Ja	Ja
$ zusammenführen	-	-	Nein	Nein

Befehl	3.6	4,0	5.0	Elastischer Cluster
$out	Ja	Ja	Ja	Nein
$planCacheStats	-	-	Nein	Nein
$project	Ja	Ja	Ja	Ja
$redact	Ja	Ja	Ja	Ja
$replaceRoot	Ja	Ja	Ja	Ja
$sample	Ja	Ja	Ja	Ja
$set	-	-	Nein	Nein
$setWindowFields	-	-	Nein	Nein
$skip	Ja	Ja	Ja	Ja
$sort	Ja	Ja	Ja	Ja
$sortByCount	Nein	Nein	Nein	Nein
$ Union mit	-	-	Nein	Nein
$unset	-	-	Nein	Nein
$unwind	Ja	Ja	Ja	Ja

Zeichenfolgenoperatoren

Befehl	3.6	4,0	5.0	Elastischer Cluster
$concat	Ja	Ja	Ja	Ja
$indexOfBytes	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
$indexOfCP	Ja	Ja	Ja	Ja
$ltrim	Nein	Nein	Nein	Nein
$RegexFind	-	-	Ja	Nein
$regexFindAll	-	-	Nein	Nein
$regexMatch	-	-	Ja	Nein
$ Alles ersetzen	-	-	Nein	Nein
$ ersetzenEins	-	-	Nein	Nein
$rtrim	Nein	Nein	Nein	Nein
$split	Ja	Ja	Ja	Ja
$strcasecmp	Ja	Ja	Ja	Ja
$strLenBytes	Ja	Ja	Ja	Ja
$strLenCP	Ja	Ja	Ja	Ja
$substr	Ja	Ja	Ja	Ja
$substrBytes	Ja	Ja	Ja	Ja
$substrCP	Ja	Ja	Ja	Ja
$toLower	Ja	Ja	Ja	Ja
$toUpper	Ja	Ja	Ja	Ja
$ trimmen	Nein	Nein	Nein	Nein

Systemvariablen

Befehl	3.6	4,0	5.0	Elastischer Cluster
$$CURRENT	Nein	Nein	Nein	Nein
$$DESCEND	Ja	Ja	Ja	Ja
$$KEEP	Ja	Ja	Ja	Ja
$$PRUNE	Ja	Ja	Ja	Ja
$$REMOVE	Nein	Nein	Nein	Nein
$$ROOT	Ja	Ja	Ja	Ja

Operator für die Textsuche

Befehl	3.6	4,0	5.0	Elastischer Cluster
$meta	Nein	Nein	Ja	Nein
$search	Nein	Nein	Ja	Nein

Geben Sie Konvertierungsoperatoren ein

Befehl	3.6	4,0	5.0	Elastischer Cluster
$ konvertieren	Nein	Ja	Ja	Ja
$ ist Zahl	-	-	Nein	Nein
$ zu Bool	Nein	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
$ bis heute	Nein	Ja	Ja	Ja
$ bis Dezimal	Nein	Ja	Ja	Ja
$toDouble	Nein	Ja	Ja	Ja
$ nach Int	Nein	Ja	Ja	Ja
$ zu lang	Nein	Ja	Ja	Ja
$toObjectId	Nein	Ja	Ja	Ja
$ toString	Nein	Ja	Ja	Ja

Operatoren für Variablen

Befehl	3.6	4,0	5.0	Elastischer Cluster
$let	Ja	Ja	Ja	Ja
$map	Ja	Ja	Ja	Ja

Verschiedene Betreiber

Befehl	3.6	4,0	5.0	Elastischer Cluster
$getField	-	-	Nein	Nein
$ Rand	-	-	Nein	Nein
$Abtastrate	-	-	Nein	Nein

Datentypen

Befehl	3.6	4,0	5.0	Elastischer Cluster
32-Bit-Ganzzahl (int)	Ja	Ja	Ja	Ja
64-Bit-Ganzzahl (lang)	Ja	Ja	Ja	Ja
Array	Ja	Ja	Ja	Ja
Binäre Daten	Ja	Ja	Ja	Ja
Boolesch	Ja	Ja	Ja	Ja
Datum	Ja	Ja	Ja	Ja
DBPointer	Nein	Nein	Nein	Nein
DBRefs	Nein	Nein	Nein	Nein
Decimal128	Ja	Ja	Ja	Ja
Double	Ja	Ja	Ja	Ja
JavaScript	Nein	Nein	Nein	Nein
JavaScript(mit Geltungsbereich)	Nein	Nein	Nein	Nein
MaxKey	Ja	Ja	Ja	Ja
MinKey	Ja	Ja	Ja	Ja
Null	Ja	Ja	Ja	Ja
Object	Ja	Ja	Ja	Ja
ObjectId	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
Regulärer Ausdruck	Ja	Ja	Ja	Ja
String	Ja	Ja	Ja	Ja
Symbol	Nein	Nein	Nein	Nein
Zeitstempel	Ja	Ja	Ja	Ja
Undefined	Nein	Nein	Nein	Nein

Indizes und Indexeigenschaften

Themen

- Indizes
- Eigenschaften indizieren

Indizes

Befehl	3.6	4,0	5.0	Elastischer Cluster
2dsphere	Ja	Ja	Ja	Ja
2D-Index	Nein	Nein	Nein	Nein
Verbundindex	Ja	Ja	Ja	Ja
Hash-Index	Nein	Nein	Nein	Nein
Multikey-Index	Ja	Ja	Ja	Ja
Einzelfeldindex	Ja	Ja	Ja	Ja

Befehl	3.6	4,0	5.0	Elastischer Cluster
Textindex	Nein	Nein	Ja	Nein
Platzhalter	Nein	Nein	Nein	Nein

Eigenschaften indizieren

Befehl	3.6	4,0	5.0	Elastischer Cluster
Hintergrund	Ja	Ja	Ja	Ja
Berücksichtigt Groß- und Kleinschreibung nicht	Nein	Nein	Nein	Nein
Hidden	Nein	Nein	Nein	Nein
Teilweise	Nein	Nein	Ja	Nein
Sparse	Ja	Ja	Ja	Ja
Text	Nein	Nein	Ja	Nein
TTL	Ja	Ja	Ja	Ja
Eindeutig	Ja	Ja	Ja	Ja
Vektor	Nein	Nein	Ja	Nein

Generative künstliche Intelligenz von Amazon DocumentDB

Amazon DocumentDB bietet Funktionen, mit denen Modelle für maschinelles Lernen (ML) und generative künstliche Intelligenz (KI) in Echtzeit mit in Amazon DocumentDB gespeicherten Daten arbeiten können. Kunden müssen keine Zeit mehr damit verbringen, separate Infrastrukturen zu verwalten, Code für die Verbindung mit einem anderen Service zu schreiben und Daten aus ihrer Primärdatenbank zu duplizieren.

Weitere Informationen zu künstlicher Intelligenz und dazu, wie Sie Ihre KI-Anforderungen unterstützen AWS können, finden Sie in diesem „Was ist“ -Artikel.

Themen

- Maschinelles Lernen ohne Code mit Amazon SageMaker AI Canvas
- Vektorsuche für Amazon DocumentDB

Maschinelles Lernen ohne Code mit Amazon SageMaker AI Canvas

Mit Amazon SageMaker AI Canvas können Sie Ihre eigenen KI/ML-Modelle erstellen, ohne eine einzige Codezeile schreiben zu müssen. Sie können ML-Modelle für gängige Anwendungsfälle wie Regression und Prognose erstellen und auf Fundamentmodelle (FMs) von Amazon Bedrock zugreifen und diese auswerten. Sie können auch FMs von Amazon SageMaker AI aus auf public zugreifen, JumpStart um Inhalte zu generieren, Text zu extrahieren und Text zusammenzufassen, um generative KI-Lösungen zu unterstützen.

Wie erstellt man ML-Modelle ohne Code mit AI Canvas SageMaker

Amazon DocumentDB ist jetzt in Amazon SageMaker AI Canvas integriert, um maschinelles Lernen (ML) ohne Code mit in Amazon DocumentDB gespeicherten Daten zu ermöglichen. Sie können jetzt ML-Modelle für Regressions- und Prognoseanforderungen erstellen und Basismodelle für die Zusammenfassung und Generierung von Inhalten verwenden, indem Sie in Amazon DocumentDB gespeicherte Daten verwenden, ohne eine einzige Codezeile schreiben zu müssen.

SageMaker AI Canvas bietet eine visuelle Oberfläche, mit der Amazon DocumentDB-Kunden Prognosen erstellen können, ohne dass KI/ML-Kenntnisse erforderlich sind oder eine einzige Codezeile geschrieben werden muss. Kunden können jetzt den SageMaker AI Canvas-

Workspace aus den AWS Management Console Amazon DocumentDB DocumentDB-Daten für die Datenvorbereitung und das Modelltraining starten, importieren und verknüpfen. Daten in Amazon DocumentDB können jetzt in SageMaker AI Canvas verwendet werden, um Modelle zur Vorhersage der Kundenabwanderung, zur Aufdeckung von Betrug, zur Vorhersage von Wartungsausfällen, zur Prognose von Geschäftskennzahlen und zur Generierung von Inhalten zu erstellen und zu erweitern. Kunden können nun mithilfe der systemeigenen Integration von SageMaker AI Canvas ML-gestützte Erkenntnisse veröffentlichen und teamübergreifend teilen. QuickSight Datenerfassungspipelines in SageMaker AI Canvas werden standardmäßig auf sekundären Amazon DocumentDB DocumentDB-Instances ausgeführt, wodurch sichergestellt wird, dass die Leistung von Anwendungs- und SageMaker AI Canvas-Erfassungs-Workloads nicht beeinträchtigt wird.

Amazon DocumentDB-Kunden können mit SageMaker AI Canvas beginnen, indem sie zur neuen Amazon DocumentDB No-Code ML Console-Seite navigieren und sich mit neuen oder verfügbaren SageMaker AI Canvas-Workspaces verbinden.

Konfiguration der AI-Domäne und des Benutzerprofils SageMaker

Sie können von SageMaker AI-Domänen aus, die im Modus Nur VPC ausgeführt werden, eine Verbindung zu Amazon DocumentDB-Clustern herstellen. Durch den Start einer SageMaker KI-Domain in Ihrer VPC können Sie den Datenfluss von Ihren SageMaker AI Studio- und Canvas-Umgebungen aus steuern. Auf diese Weise können Sie den Internetzugang einschränken, den Datenverkehr mithilfe standardmäßiger AWS Netzwerk- und Sicherheitsfunktionen überwachen und überprüfen und über VPC-Endpunkte eine Verbindung zu anderen AWS Ressourcen herstellen. Informationen zur Erstellung Ihrer SageMaker KI-Domain für die Verbindung mit Ihrem Amazon DocumentDB-Cluster finden Sie unter Amazon SageMaker AI Canvas Getting started und Configure Amazon SageMaker AI Canvas in a VPC without internet access im Amazon SageMaker AI Developer Guide.

Konfiguration von IAM-Zugriffsberechtigungen für Amazon DocumentDB und AI Canvas SageMaker

Ein Amazon DocumentDB DocumentDB-Benutzer, `AmazonDocDBConsoleFullAccess` der seiner zugehörigen Rolle und Identität zugeordnet ist, kann auf die AWS Management Console zugreifen. Fügen Sie der oben genannten Rolle oder Identität die folgenden Aktionen hinzu, um Zugriff auf maschinelles Lernen ohne Code mit Amazon SageMaker AI Canvas zu gewähren.

```
"sagemaker:CreatePresignedDomainUrl",
"sagemaker:DescribeDomain",
```

```
"sagemaker:ListDomains",
"sagemaker:ListUserProfiles"
```

Datenbankbenutzer und Rollen für SageMaker AI Canvas erstellen

Mithilfe der rollenbasierten Zugriffskontrolle (RBAC) in Amazon DocumentDB können Sie den Zugriff auf die Aktionen einschränken, die Benutzer an Datenbanken ausführen können. Bei RBAC werden einem Benutzer eine oder mehrere Rollen gewährt. Diese Rollen bestimmen die Operationen, die ein Benutzer mit Datenbankressourcen ausführen kann.

Als Canvas-Benutzer stellen Sie mit Benutzernamen und Passwort eine Verbindung zu einer Amazon DocumentDB DocumentDB-Datenbank her. Sie können mithilfe der Amazon DocumentDB RBAC-Funktionalität einen Datenbankbenutzer/eine Datenbankrolle für einen Canvas-Benutzer erstellen, der Lesezugriff auf die spezifischen Datenbanken hat.

Verwenden Sie zum Beispiel die Operation: `createUser`

```
db.createUser({
user: "canvas_user",
pwd: "<insert-password>",
roles: [{role: "read", db: "sample-database-1"}]
})
```

Dadurch wird eine erstellt`canvas_user`, die über Leseberechtigungen für die `sample-database-1` Datenbank verfügt. Ihre Canvas-Analysten können diese Anmeldeinformationen verwenden, um auf Daten in Ihrem Amazon DocumentDB-Cluster zuzugreifen. Weitere Informationen finden Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle Sie unter.

Verfügbare Regionen

Die No-Code-Integration ist in Regionen verfügbar, in denen sowohl Amazon DocumentDB als auch Amazon SageMaker AI Canvas unterstützt werden. Zu den Regionen gehören:

- us-east-1 (N. Virginia)
- us-east-2 (Ohio)
- us-west-2 (Oregon)
- ap-northeast-1 (Tokyo)
- ap-northeast-2 (Seoul)

- ap-south-1 (Mumbai)
- ap-southeast-1 (Singapur)
- ap-southeast-2 (Sydney)
- eu-central-1 (Frankfurt)
- eu-west-1 (Ireland)

Die aktuelle Verfügbarkeit in der Region finden Sie unter Amazon SageMaker SageMaker AI Canvas im Amazon AI Developer Guide.

Vektorsuche für Amazon DocumentDB

Die Vektorsuche ist eine Methode, die beim maschinellen Lernen verwendet wird, um Datenpunkte zu finden, die einem bestimmten Datenpunkt ähneln, indem ihre Vektordarstellungen anhand von Entfernungs- oder Ähnlichkeitsmetriken verglichen werden. Je näher sich die beiden Vektoren im Vektorraum befinden, desto ähnlicher wird davon ausgegangen, dass sich die zugrunde liegenden Elemente ähneln. Diese Technik hilft dabei, die semantische Bedeutung der Daten zu erfassen. Dieser Ansatz ist in verschiedenen Anwendungen nützlich, z. B. in Empfehlungssystemen, Verarbeitung natürlicher Sprache und Bilderkennung.

Die Vektorsuche für Amazon DocumentDB kombiniert die Flexibilität und die umfangreichen Abfragefunktionen einer JSON-basierten Dokumentendatenbank mit der Leistungsfähigkeit der Vektorsuche. Wenn Sie Ihre vorhandenen Amazon DocumentDB-Daten oder eine flexible Dokumentendatenstruktur verwenden möchten, um Anwendungsfälle für maschinelles Lernen und generative KI wie semantische Suche, Produktempfehlungen, Personalisierung, Chatbots, Betrugserkennung und Anomalieerkennung zu erstellen, dann ist die Vektorsuche für Amazon DocumentDB die ideale Wahl für Sie. Die Vektorsuche ist auf instanzbasierten Amazon DocumentDB 5.0-Clustern verfügbar.

Themen

- Vektoren einfügen
- Einen Vektorindex erstellen
- Eine Indexdefinition abrufen
- Vektoren abfragen
- Funktionen und Einschränkungen

- Bewährte Methoden

Vektoren einfügen

Um Vektoren in Ihre Amazon DocumentDB DocumentDB-Datenbank einzufügen, können Sie bestehende Einfügemethoden verwenden:

Beispiel

Im folgenden Beispiel wird eine Sammlung von fünf Dokumenten in einer Testdatenbank erstellt. Jedes Dokument enthält zwei Felder: den Produktnamen und die entsprechende Vektoreinbettung.

```
db.collection.insertMany([
  {"product_name": "Product A", "vectorEmbedding": [0.2, 0.5, 0.8]},
  {"product_name": "Product B", "vectorEmbedding": [0.7, 0.3, 0.9]},
  {"product_name": "Product C", "vectorEmbedding": [0.1, 0.2, 0.5]},
  {"product_name": "Product D", "vectorEmbedding": [0.9, 0.6, 0.4]},
  {"product_name": "Product E", "vectorEmbedding": [0.4, 0.7, 0.2]}
]);
```

Einen Vektorindex erstellen

Amazon DocumentDB unterstützt sowohl die Indexierung Hierarchical Navigable Small World (HNSW) als auch die Indexierungsmethoden Inverted File with Flat Compression (). IVFFlat Ein IVFFlat Index unterteilt Vektoren in Listen und durchsucht anschließend eine ausgewählte Teilmenge der Listen, die dem Abfragevektor am nächsten sind. Andererseits organisiert ein HNSW-Index die Vektordaten in einem vielschichtigen Diagramm. HNSW hat zwar im Vergleich zu langsamere Erstellungszeiten IVFFlat, bietet aber eine bessere Abfrageleistung und einen besseren Abruf. Im IVFFlat Gegensatz zu HNSW ist kein Trainingsschritt erforderlich, sodass der Index ohne anfängliches Laden von Daten generiert werden kann. Für die meisten Anwendungsfälle empfehlen wir, den HNSW-Indextyp für die Vektorsuche zu verwenden.

Wenn Sie keinen Vektorindex erstellen, führt Amazon DocumentDB eine exakte Suche nach dem nächsten Nachbarn durch und gewährleistet so einen perfekten Wiedererkennungswert. In Produktionsszenarien ist Geschwindigkeit jedoch entscheidend. Wir empfehlen die Verwendung von Vektorindizes, bei denen ein gewisser Rückruf gegen eine höhere Geschwindigkeit eingetauscht werden kann. Es ist wichtig zu beachten, dass das Hinzufügen eines Vektorindexes zu unterschiedlichen Abfrageergebnissen führen kann.

Vorlagen

Sie können die folgenden `createIndex` oder `runCommand` Vorlagen verwenden, um einen Vektorindex für ein Vektorfeld zu erstellen:

Using createIndex

In bestimmten Treibern, wie z. B. Mongosh und Java, `createIndex` kann die Verwendung der `vectorOptions` Parameter in zu einem Fehler führen. In solchen Fällen empfehlen wir die Verwendung von`runCommand`:

```
db.collection.createIndex(
  { "<vectorField>": "vector" },
  { "name": "<indexName>",
    "vectorOptions": {
      "type": " <hnsw> | <ivfflat> ",
      "dimensions": <number_of_dimensions>,
      "similarity": " <euclidean> | <cosine> | <dotProduct> ",
      "lists": <number_of_lists> [applicable for IVFFlat],
      "m": <max number of connections> [applicable for HNSW],
      "efConstruction": <size of the dynamic list for index build> [applicable for
 HNSW]
    }
  }
);
```

Using runCommand

In bestimmten Treibern wie Mongosh und Java `createIndex` kann die Verwendung der `vectorOptions` Parameter in zu einem Fehler führen. In solchen Fällen empfehlen wir die Verwendung von`runCommand`:

```
db.runCommand(
  { "createIndexes": "<collection>",
  "indexes": [{
      key: { "<vectorField>": "vector" },
      vectorOptions: {
          type: " <hnsw> | <ivfflat> ",
          dimensions: <number of dimensions>,
          similarity: " <euclidean> | <cosine> | <dotProduct> ",
          lists: <number_of_lists> [applicable for IVFFlat],
          m: <max number of connections> [applicable for HNSW],
          efConstruction: <size of the dynamic list for index build> [applicable for
 HNSW]
```

```
            },
        name: "myIndex"
        }]
    }
);
```

Parameter	Anforderung	Datentyp	Beschreibung	Wert (e)
name	optional	Zeichenfolge	Gibt den Namen des Indexes an.	Alphanumerisch
type	optional		Gibt den Indextyp an.	Unterstützt: hnsw oder ivfflat Standard: HNSW (Engine-Patch 3.0.4574 und höher)
dimensions	Erforderlich	Ganzzahl	Gibt die Anzahl der Dimensionen in den Vektordaten an.	Maximal 2.000 Dimensionen.
similarity	Erforderlich	Zeichenfolge	Gibt die Entfernungsmetrik an, die für die Ähnlichkeitsberechnung verwendet wird.	• **euclidean** • **cosine** • **dotProduct**
lists	erforderlich für IVFFlat	Ganzzahl	Gibt die Anzahl der Cluster an, die der IVFFlat Index verwendet, um die Vektordaten zu gruppieren. Die empfohlen	Minimum: 1 Maximum: Weitere Informationen finden Sie in der folgenden Tabelle mit

Parameter	Anforderung	Datentyp	Beschreibung	Wert (e)
			e Einstellung ist die Anzahl der Dokumente /1000 für bis zu 1 Mio. Dokumente und `sqrt(# of documents)` für über 1 Mio. Dokumente.	den Listen pro Instanzty p. Funktionen und Einschrän kungen
`m`	optional	Ganzzahl	Gibt die maximale Anzahl von Verbindun gen für einen HNSW-Index an	Standard: 16 Bereich [2, 100]
`efConstru ction`	optional	Ganzzahl	Gibt die Größe der dynamischen Kandidatenliste für die Erstellung des Graphen für den HNSW-Inde x an. `efConstru ction` muss größer oder gleich (2 * m) sein	Standard: 64 Bereich [4, 1000]

Es ist wichtig, dass Sie den Wert von Unterparametern wie `lists` für IVFFlat und `m` und `efConstruction` für HNSW entsprechend festlegen, da sich dies auf die Genauigkeit/den Abruf, die Erstellungszeit und die Leistung Ihrer Suche auswirkt. Ein höherer Listenwert erhöht die Geschwindigkeit der Abfrage, da dadurch die Anzahl der Vektoren in jeder Liste reduziert wird, was zu kleineren Regionen führt. Eine kleinere Regionsgröße kann jedoch zu mehr Erinnerungsfehlern

führen, was wiederum zu einer geringeren Genauigkeit führt. Bei HNSW `efConstruction` erhöht sich durch die Erhöhung des Werts von `m` und die Genauigkeit, aber auch die Zeit und Größe der Indexerstellung. Im Folgenden sind einige Beispiele aufgeführt:

Beispiele

HNSW

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "hnsw",
      "dimensions": 3,
      "similarity": "euclidean",
      "m": 16,
      "efConstruction": 64
    }
  }
);
```

IVFFlat

```
db.collection.createIndex(
  { "vectorEmbedding": "vector" },
  { "name": "myIndex",
    "vectorOptions": {
      "type": "ivfflat",
      "dimensions": 3,
      "similarity": "euclidean",
      "lists":1
    }
  }
)
```

Eine Indexdefinition abrufen

Sie können die Details Ihrer Indizes, einschließlich Vektorindizes, mit dem `getIndexes` folgenden Befehl anzeigen:

Beispiel

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

Beispielausgabe

```
[
 {
  "v" : 4,
  "key" : {
   "_id" : 1
  },
  "name" : "_id_",
  "ns" : "test.collection"
 },
 {
  "v" : 4,
  "key" : {
   "vectorEmbedding" : "vector"
  },
  "name" : "myIndex",
  "vectorOptions" : {
   "type" : "ivfflat",
   "dimensions" : 3,
   "similarity" : "euclidean",
   "lists" : 1
  },
  "ns" : "test.collection"
 }
]
```

Vektoren abfragen

Vorlage für Vektor-Abfragen

Verwenden Sie die folgende Vorlage, um einen Vektor abzufragen:

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": <query vector>,
        "path": "<vectorField>",
        "similarity": "<distance metric>",
```

```
        "k": <number of results>,
        "probes":<number of probes> [applicable for IVFFlat],
        "efSearch":<size of the dynamic list during search> [applicable for HNSW]
      }
    }
  }
]);
```

Parameter	Anforderung	Typ	Beschreibung	Wert (e)
vectorSearch	Erforderlich	operator	Wird im Befehl $search verwendet, um die Vektoren abzufragen.	
vector	Erforderlich	Array	Gibt den Abfragevektor an, der verwendet wird, um ähnliche Vektoren zu finden.	
path	Erforderlich	Zeichenfolge	Definiert den Namen des Vektorfeldes.	
k	Erforderlich	Ganzzahl	Gibt die Anzahl der Ergebnisse an, die die Suche zurückgibt.	
similarity	Erforderlich	Zeichenfolge	Gibt die Entfernungsmetrik an, die für die Ähnlichke	• **euclidean** • **cosine** • **dotProduct**

Parameter	Anforderung	Typ	Beschreibung	Wert (e)
			itsberechnung verwendet wird.	
`probes`	optional	Ganzzahl	Die Anzahl der Cluster, die bei der Vektorsuche untersucht werden sollen. Ein höherer Wert sorgt für einen besseren Abruf auf Kosten der Geschwindigkeit. Er kann auf die Anzahl der Listen für die exakte Suche nach dem nächsten Nachbarn gesetzt werden (ab diesem Zeitpunkt verwendet der Planer den Index nicht mehr). Die empfohlene Einstellung, um mit der Feinabstimmung zu beginnen, ist`sqrt(# of lists)`.	Standard: 1

Parameter	Anforderung	Typ	Beschreibung	Wert (e)
`efSearch`	optional	Ganzzahl	Gibt die Größe der dynamisch en Kandidate nliste an, die der HNSW-Index bei der Suche verwendet. Ein höherer Wert von `efSearch` ermöglicht einen besseren Abruf auf Kosten der Geschwindigkeit.	Standard: 40 Bereich [1, 1000]

Es ist wichtig, den Wert von `efSearch` (HNSW) oder `probes` (IVFFlat) genau einzustellen, um die gewünschte Leistung und Genauigkeit zu erreichen. Sehen Sie sich die folgenden Beispieloperationen an:

HNSW

```
db.collection.aggregate([
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "efSearch": 40
      }
    }
  }
]);
```

IVFFlat

```
db.collection.aggregate([
```

```
  {
    $search: {
      "vectorSearch": {
        "vector": [0.2, 0.5, 0.8],
        "path": "vectorEmbedding",
        "similarity": "euclidean",
        "k": 2,
        "probes": 1
      }
    }
  }
]);
```

Beispielausgabe

Die Ausgabe dieser Operation sieht in etwa wie folgt aus:

```
{ "_id" : ObjectId("653d835ff96bee02cad7323c"), "product_name" : "Product A",
 "vectorEmbedding" : [ 0.2, 0.5, 0.8 ] }
{ "_id" : ObjectId("653d835ff96bee02cad7323e"), "product_name" : "Product C",
 "vectorEmbedding" : [ 0.1, 0.2, 0.5 ] }
```

Funktionen und Einschränkungen

Versionskompatibilität

- Die Vektorsuche für Amazon DocumentDB ist nur auf instanzbasierten Amazon DocumentDB 5.0-Clustern verfügbar.

Vektoren

- Amazon DocumentDB kann Vektoren mit bis zu 2.000 Dimensionen indizieren. Es können jedoch bis zu 16.000 Dimensionen ohne Index gespeichert werden.

Indizes

- Für die IVFFlat Indexerstellung ist die empfohlene Einstellung für den Listenparameter die Anzahl der Dokumente/1000 für bis zu 1 Million Dokumente und `sqrt(# of documents)` für über 1 Million Dokumente. Aufgrund eines Arbeitsspeicherlimits unterstützt Amazon DocumentDB einen

bestimmten Höchstwert des Lists-Parameters, der von der Anzahl der Dimensionen abhängt. Zu Ihrer Information enthält die folgende Tabelle die Höchstwerte des Listenparameters für Vektoren mit 500, 1000 und 2.000 Dimensionen:

Instance-Typ	Listen mit 500 Dimensionen	Listen mit 1000 Dimensionen	Listen mit 2000 Dimensionen
t3.med	372	257	150
r5l	915	741	511
r5.xl	1.393	1.196	901
r5,2 xl	5.460	5.230	4.788
r5,4 xl	7.842	7.599	7.138
r5,8 xl	11.220	10.974	10.498
r5,12 xl	13.774	13.526	13.044
r5,16xl	15.943	15.694	15.208
r5,24 xl	19.585	19.335	18.845

- Keine anderen Indexoptionen wie `compound` Vektorindizes `sparse` oder `partial` werden von Vektorindizes unterstützt.
- Die parallele Indexerstellung wird für den HNSW-Index nicht unterstützt. Es wird nur für den IVFFlat Index unterstützt.

Vektor-Abfrage

- Bei der Vektor-Suchanfrage ist es wichtig, die Parameter wie `probes` oder `efSearch` für optimale Ergebnisse zu optimieren. Je höher der Wert des `efSearch` Parameters `probes` oder, desto höher der Abruf und desto niedriger die Geschwindigkeit. Die empfohlene Einstellung, um mit der Feinabstimmung des Sondenparameters zu beginnen, ist`sqrt(# of lists)`.

Bewährte Methoden

Lernen Sie bewährte Methoden für die Arbeit mit der Vektorsuche in Amazon DocumentDB kennen. Dieser Abschnitt wird fortlaufend aktualisiert, wenn neue bewährte Methoden identifiziert werden.

- Bei der Indexerstellung von Inverted File with Flat Compression (IVFFlat) werden die Datenpunkte auf der Grundlage von Ähnlichkeiten gruppiert und organisiert. Damit ein Index effektiver ist, empfehlen wir daher, vor der Indexerstellung zumindest einige Daten zu laden.
- Bei Vektor-Suchanfragen ist es wichtig, die Parameter genau einzustellen, um `efSearch` optimale Ergebnisse zu erzielen. `probes` Je höher der Wert des `efSearch` Parameters `probes` oder ist, desto höher ist der Recall und desto niedriger ist die Geschwindigkeit. Die empfohlene Einstellung, um mit der Feinabstimmung des `probes` Parameters zu beginnen, ist`sqrt(lists)`.

Ressourcen

- Vektorsuche: Was ist ein neuer Blogbeitrag
- Beispiel für einen semantischen Suchcode
- Codebeispiele für die Vektorsuche in Amazon DocumentDB

Migration zu Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist ein vollständig verwalteter Datenbankservice, der mit der MongoDB-API kompatibel ist. Sie können Ihre Daten von MongoDB-Datenbanken, die lokal oder auf Amazon Elastic Compute Cloud (Amazon EC2) laufen, nach Amazon DocumentDB migrieren, indem Sie den in diesem Abschnitt beschriebenen Prozess verwenden.

Themen

- Migrationstools
- Erkennung
- Planung: Amazon DocumentDB-Clusteranforderungen
- Migrationsansätze
- Quellen für die Migration
- Konnektivität bei der Migration
- Testen
- Leistungstests
- Failover-Tests
- Weitere Ressourcen
- Leitfaden für die Migration: MongoDB zu Amazon DocumentDB

Migrationstools

Für die Migration zu Amazon DocumentDB verwenden die meisten Kunden hauptsächlich die beiden Tools AWS Database Migration Service (AWS DMS) und Befehlszeilenprogramme wie `mongodump` und`mongorestore`. Als bewährte Methode und für jede dieser Optionen empfehlen wir, dass Sie zuerst Indizes in Amazon DocumentDB erstellen, bevor Sie mit der Migration beginnen, da dies die Gesamtzeit reduzieren und die Geschwindigkeit der Migration erhöhen kann. Dazu können Sie das Amazon DocumentDB Index Tool verwenden.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) ist ein Cloud-Service, der es einfach macht, relationale Datenbanken und nicht-relationale Datenbanken zu Amazon DocumentDB zu migrieren. Sie können AWS DMS es verwenden, um Ihre Daten von lokal oder lokal gehosteten Datenbanken zu Amazon

DocumentDB zu migrieren. EC2 Mit AWS DMS können Sie einmalige Migrationen durchführen oder laufende Änderungen replizieren, um Quellen und Ziele synchron zu halten.

Weitere Informationen zur Migration AWS DMS zu Amazon DocumentDB finden Sie unter:

- MongoDB als Quelle verwenden für AWS DMS
- Verwenden von Amazon DocumentDB als Ziel für AWS Database Migration Service
- Exemplarische Vorgehensweise: Migration von MongoDB zu Amazon DocumentDB

Befehlszeilen-Dienstprogramme

Zu den gängigen Hilfsprogrammen für die Migration von Daten zu und von Amazon DocumentDB gehören`mongodump`, `mongorestore``mongoexport`, und. `mongoimport` In der Regel sind `mongodump` und `mongorestore` die effizientesten Dienstprogramme, da sie Daten aus Ihren Datenbanken in einem binären Format sichern und wiederherstellen. Dies ist im Allgemeinen die beste Option und führt im Vergleich zu logischen Exporten zu einer geringeren Datengröße. `mongoexport`und `mongoimport` sind nützlich, wenn Sie Daten in einem logischen Format wie JSON oder CSV exportieren und importieren möchten, da die Daten für Menschen lesbar sind, aber im Allgemeinen langsamer sind als `mongodump` `mongorestore` /und eine größere Datengröße ergeben.

Im folgenden Migrationsansätze Abschnitt wird erläutert, wann die Verwendung AWS DMS von Befehlszeilenprogrammen am besten ist, je nach Anwendungsfall und Anforderungen.

Erkennung

Für jede Ihrer MongoDB-Bereitstellungen sollten Sie zwei Datasets identifizieren und protokollieren: Architekturdetails und Betriebseigenschaften. Diese Informationen helfen Ihnen bei der Auswahl des geeigneten Migrationsansatzes und der Clustergröße.

Architekturdetails

- Name

  Wählen Sie einen eindeutigen Namen für die Verfolgung dieser Bereitstellung aus.

- Version

Erfassen Sie die Version von MongoDB, die Ihre Bereitstellung ausführt. Um die Version zu finden, verbinden Sie sich über die Mongo-Shell mit einem Mitglied des Replikatsatzes und führen die Operation `db.version()` aus.

- Typ

  Protokollieren Sie, ob es sich bei Ihrer Bereitstellung um eine eigenständige Mongo-Instance, einen Replikatsatz oder einen Sharded-Cluster handelt.

- Mitglieder

  Protokollieren Sie die Hostnamen, Adressen und Ports der einzelnen Cluster, Replikatsätze oder eigenständigen Mitglieder.

  Für eine geclusterte Bereitstellung können Sie Shard-Mitglieder finden, indem Sie sich per Mongo-Shell mit einem Mongo-Host verbinden und die Operation `sh.status()` ausführen.

  Sie können die Mitglieder eines Replikatsatzes abrufen, indem Sie sich per Mongo-Shell mit einem Replikatsatz verbinden und die Operation `rs.status()` ausführen.

- Oplog-Größen

  Bei Replikatsätzen oder Sharded-Clustern notieren Sie die Größe des Oplogs für jedes Replikatsatzmitglied. Um die Oplog-Größe eines Mitglieds zu finden, verbinden Sie sich mit dem Replikat-Set-Element mit der Mongo-Shell und führen Sie die Operation `ps.printReplicationInfo()` aus.

- Prioritäten für Replikatsatzmitglieder

  Protokollieren Sie bei den Replikatsätzen oder Sharded-Clustern die Priorität der einzelnen Replikatsatzmitglieder. Um die Prioritäten für das Replikatsatzmitglied zu finden, verbinden Sie sich

per Mongo-Shell mit einem Replikatsatzmitglied und führen Sie die Operation `rs.conf()` aus. Die Priorität wird als Wert des Schlüssels `priority` angezeigt.

- TLS/SSL-Nutzung

  Protokollieren Sie, ob auf den einzelnen Knoten während der Übertragung TLS/SSL (Transport Layer Security/Secure Sockets Layer) verwendet wird.

Betriebseigenschaften

- Datenbankstatistik

  Protokollieren Sie für jede Sammlung die folgenden Informationen:

  - Name
  - Datengröße
  - Sammlungsanzahl

  Um die Datenbankstatistiken zu finden, verbinden Sie sich per Mongo-Shell mit Ihrer Datenbank und führen Sie den Befehl `db.runCommand({dbstats: 1})` aus.

- Sammlungsstatistik

  Protokollieren Sie für jede Sammlung die folgenden Informationen:

  - Namespace
  - Datengröße
  - Indexanzahl
  - Ob die Sammlung gedeckelt ist

- Indexstatistik

  Erfassen Sie für jede Sammlung die folgenden Indexinformationen:

  - Namespace

- ID
- Größe
- Schlüssel
- TTL
- Sparse
- Hintergrund

Um die Indexinformationen zu finden, verbinden Sie sich per Mongo-Shell mit Ihrer Datenbank und führen Sie den Befehl `db.collection.getIndexes()` aus.

- Opcounters

  Diese Informationen helfen Ihnen, Ihre aktuellen MongoDB-Workload-Muster (viele Lesevorgänge, viele Schreibvorgänge oder ausgeglichen) zu ermitteln. Es enthält auch Hinweise zur ersten Auswahl Ihrer Amazon DocumentDB DocumentDB-Instance.

  Nachfolgend sind die wichtigsten Informationen aufgeführt, die während des Überwachungszeitraums (in der Form Anzahl/Sekunde) gesammelt werden müssen:

  - Abfragen
  - Einfügungen
  - Aktualisierungen
  - Löschvorgänge

  Sie können diese Informationen abrufen, indem Sie die Ausgabe des Befehls `db.serverStatus()` über den gewünschten Zeitraum grafisch darstellen. Sie können außerdem das Mongostat-Toolverwenden, um direkt Werte für diese Statistiken zu erhalten. Mit dieser Option laufen Sie jedoch Gefahr, Ihre Migration auf Basis von Nutzungszeiträumen zu planen, die keine Spitzenauslastung darstellen.

- Netzwerkstatistik

  Diese Informationen helfen Ihnen, Ihre aktuellen MongoDB-Workload-Muster (viele Lesevorgänge, viele Schreibvorgänge oder ausgeglichen) zu ermitteln. Es enthält auch Hinweise zur ersten Auswahl Ihrer Amazon DocumentDB DocumentDB-Instance.

  Nachfolgend sind die wichtigsten Informationen aufgeführt, die während des Überwachungszeitraums (in der Form Anzahl/Sekunde) gesammelt werden müssen:

  - Verbindungen
  - Netzwerk-Bytes eingehend
  - Netzwerk-Bytes ausgehend

  Sie können diese Informationen abrufen, indem Sie die Ausgabe des Befehls `db.serverStatus()` über den gewünschten Zeitraum grafisch darstellen. Sie können außerdem das Mongostat-Toolverwenden, um direkt Werte für diese Statistiken zu erhalten. Mit dieser Option laufen Sie jedoch Gefahr, Ihre Migration auf Basis von Nutzungszeiträumen zu planen, die keine Spitzenauslastung darstellen.

Planung: Amazon DocumentDB-Clusteranforderungen

Für eine erfolgreiche Migration müssen Sie sowohl die Konfiguration Ihres Amazon DocumentDB-Clusters als auch die Art und Weise, wie Anwendungen auf Ihren Cluster zugreifen, sorgfältig abwägen. Berücksichtigen Sie bei der Ermittlung Ihrer Cluster-Anforderungen die folgenden Dimensionen:

- Verfügbarkeit

  Amazon DocumentDB bietet hohe Verfügbarkeit durch die Bereitstellung von Replikat-Instances, die in einem als Failover bezeichneten Prozess zu einer primären Instance heraufgestuft werden können. Durch die Bereitstellung von Replikations-Instances in verschiedenen Availability Zones können Sie eine höhere Verfügbarkeit erreichen.

Die folgende Tabelle enthält Richtlinien für Amazon DocumentDB DocumentDB-Bereitstellungskonfigurationen zur Erreichung bestimmter Verfügbarkeitsziele.

Verfügbarkeitsziel	Gesamtzahl der Instances	Replikas	Availability Zones
99 %	1	0	1
99,9 %	2	1	2
99,99 %	3	2	3

Für die allgemeine Systemzuverlässigkeit müssen alle Komponenten berücksichtigt werden – nicht nur die Datenbank. Bewährte Verfahren und Empfehlungen zur Erfüllung der allgemeinen Anforderungen an die Systemzuverlässigkeit finden Sie im AWS Well-Architected Reliability Pillar Whitepaper.

- Leistung

  Amazon DocumentDB DocumentDB-Instances ermöglichen es Ihnen, vom Speichervolume Ihres Clusters zu lesen und darauf zu schreiben. Es gibt unterschiedliche Typen von Cluster-Instances mit verschiedenen Speicher- und vCPU-Konfigurationen, die sich auf die Lese- und Schreibleistung Ihres Clusters auswirken. Wählen Sie anhand der in der Ermittlungsphase gesammelten Informationen einen Instance-Typ aus, der Ihre Anforderungen an die Workload-Performance abdeckt. Eine Liste mit unterstützten Instance-Typen finden Sie unter Instanzklassen verwalten.

  Bei der Auswahl eines Instance-Typs für Ihren Amazon DocumentDB-Cluster sollten Sie die folgenden Aspekte der Leistungsanforderungen Ihres Workloads berücksichtigen:

- v CPUs —Architekturen, die eine höhere Verbindungsanzahl erfordern, könnten von Instances mit mehr vCPUs profitieren.

- Arbeitsspeicher — Wenn möglich, wird die maximale Leistung erzielt, wenn Sie Ihren Arbeitsdatensatz im Arbeitsspeicher behalten. Eine erste Richtlinie besteht darin, ein Drittel des Speichers Ihrer Instance für die Amazon DocumentDB DocumentDB-Engine zu reservieren, sodass zwei Drittel für Ihren Arbeitsdatensatz übrig bleiben.

- Verbindungen — Die optimale Mindestanzahl an Verbindungen beträgt acht Verbindungen pro vCPU der Amazon DocumentDB DocumentDB-Instance. Obwohl das Verbindungslimit für Amazon DocumentDB DocumentDB-Instances viel höher ist, sinken die Leistungsvorteile zusätzlicher Verbindungen bei mehr als acht Verbindungen pro vCPU.

- Netzwerk — Bei Workloads mit einer großen Anzahl von Clients oder Verbindungen sollte die gesamte Netzwerkleistung berücksichtigt werden, die für eingefügte und abgerufene Daten erforderlich ist. Massenoperationen können die Netzwerkressourcen effizienter nutzen.

- Leistung beim Einfügen — Einfügungen einzelner Dokumente sind im Allgemeinen die langsamste Methode, Daten in Amazon DocumentDB einzufügen. Masseneinfügeoperationen können wesentlich schneller sein.

- Leseleistung — Lesevorgänge aus dem Arbeitsspeicher sind immer schneller als Lesevorgänge aus dem Speichervolume. Daher ist es ideal, die Größe des Instance-Arbeitsspeichers zu optimieren, um das Arbeits-Dataset im Arbeitsspeicher zu halten.

Amazon DocumentDB-Cluster verarbeiten nicht nur Lesevorgänge von Ihrer primären Instance, sondern werden auch automatisch als Replikatsätze konfiguriert. Sie können dann schreibgeschützte Abfragen an Lesereplikate weiterleiten, indem Sie die Leseeinstellung in Ihrem MongoDB-Treiber festlegen. Sie können den Leseverkehr skalieren, indem Sie Replikate hinzufügen, was die Gesamtbelastung der primären Instance reduziert.

Es ist möglich, Amazon DocumentDB DocumentDB-Replikate verschiedener Instance-Typen im selben Cluster bereitzustellen. Ein exemplarischer Anwendungsfall könnte sein, eine Replik mit einem größeren Instance-Typ zu erstellen, um temporären Analyse-Datenverkehr zu verarbeiten. Wenn Sie einen gemischten Satz von Instance-Typen bereitstellen, sorgen Sie dafür, dass die Failover-Priorität für jede Instance konfiguriert ist. Dadurch wird sichergestellt, dass ein Failover-Ereignis stets ein Replikat ausreichender Größe für Ihre Schreiblast hochstuft.

- Wiederherstellung

  Amazon DocumentDB sichert Ihre Daten kontinuierlich, während sie geschrieben werden. Es bietet point-in-time Wiederherstellungsfunktionen (PITR) innerhalb eines konfigurierbaren Zeitraums von 1—35 Tagen, der sogenannten Aufbewahrungsfrist für Backups. Die standardmäßige Aufbewahrungsfrist für Backups beträgt einen Tag. Amazon DocumentDB erstellt außerdem automatisch tägliche Snapshots Ihres Speichervolumens, die ebenfalls für den konfigurierten Aufbewahrungszeitraum für Backups aufbewahrt werden.

  Wenn Sie Snapshots über den Aufbewahrungszeitraum der Backups hinaus aufbewahren möchten, können Sie mit dem und () auch jederzeit manuelle Snapshots initiieren. AWS Management Console AWS Command Line Interface AWS CLI Weitere Informationen finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.

  Beachten Sie bei der Planung Ihrer Migration Folgendes:

  - Wählen Sie einen Aufbewahrungszeitraum für Backups von 1—35 Tagen, der Ihrem Recovery Point Objective (RPO) entspricht.
  - Entscheiden Sie, ob und in welchem Intervall Sie manuelle Snapshots benötigen.

Migrationsansätze

Es gibt drei Hauptansätze für die Migration Ihrer Daten zu Amazon DocumentDB.

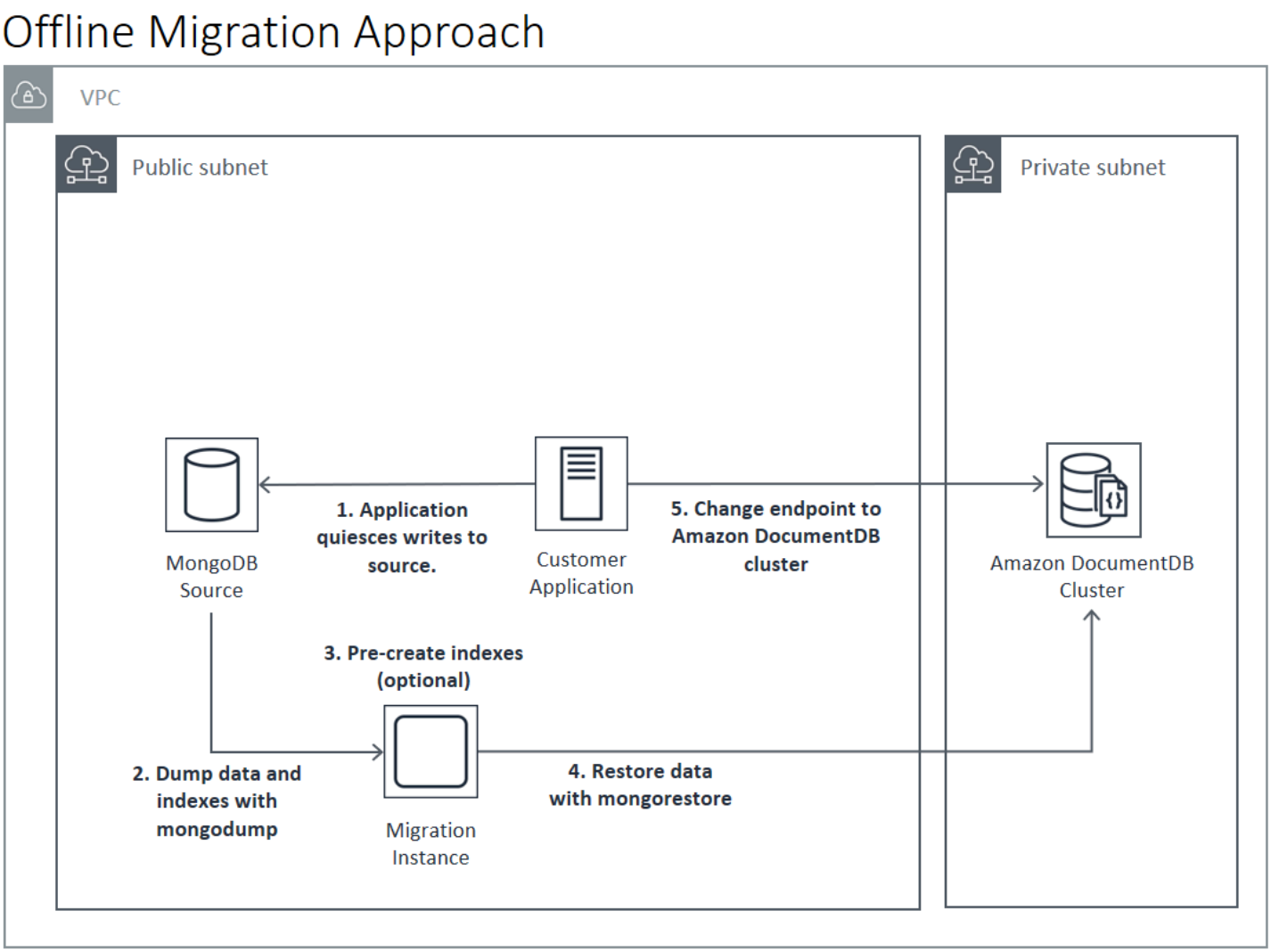
> **Note**
>
> Obwohl Sie in Amazon DocumentDB jederzeit Indizes erstellen können, ist es insgesamt schneller, Ihre Indizes zu erstellen, bevor Sie große Datensätze importieren. Als bewährte Methode empfehlen wir, dass Sie für jeden der unten aufgeführten Ansätze zuerst Ihre Indizes in Amazon DocumentDB erstellen, bevor Sie die Migration durchführen. Dazu können Sie das Amazon DocumentDB Index Tool verwenden.

Offline

Der Offline-Ansatz verwendet die `mongorestore` Tools `mongodump` und, um Ihre Daten von Ihrer MongoDB-Quellbereitstellung zu Ihrem Amazon DocumentDB-Cluster zu migrieren. Die Offline-Methode ist der einfachste Migrationsansatz, hat jedoch auch die längste Ausfallzeit für Ihren Cluster zur Folge.

Die grundlegende Vorgehensweise bei der Offline-Migration sieht wie folgt aus:

1. Stilllegen von Schreibvorgängen in Ihrer MongoDB-Quelle
2. Dump von Sammlungsdaten und Indizes aus der MongoDB-Quellenbereitstellung
3. Wenn Sie zu einem Elastic Cluster migrieren, erstellen Sie Ihre Sharded-Sammlungen mithilfe des Befehls. `sh.shardCollection()` Wenn Sie zu einem instanzbasierten Cluster migrieren, fahren Sie mit dem nächsten Schritt fort.
4. Stellen Sie Indizes im Amazon DocumentDB-Cluster wieder her.
5. Stellen Sie die Sammlungsdaten im Amazon DocumentDB-Cluster wieder her.
6. Ändern Sie Ihren Anwendungsendpunkt so, dass er in den Amazon DocumentDB-Cluster schreibt.



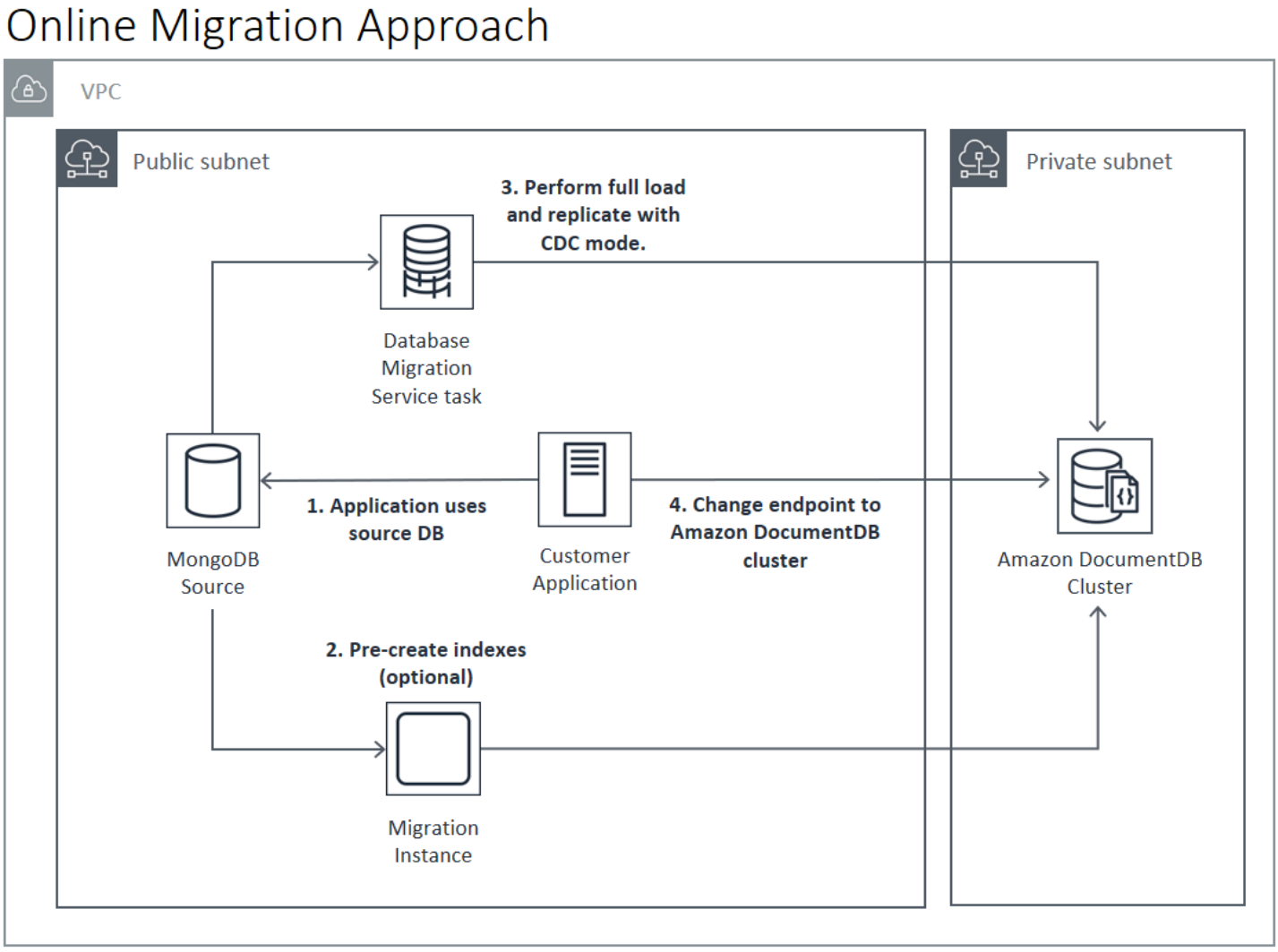
Status "Online"

Der Online- Ansatz verwendet AWS Database Migration Service (AWS DMS). Es führt ein vollständiges Laden von Daten aus Ihrer MongoDB-Quellbereitstellung in Ihren Amazon DocumentDB-Cluster durch. Danach findet ein Wechsel zum CDC-Modus (Change Data Capture) zum Replizieren von Änderungen statt. Der Onlineansatz minimiert die Ausfallzeiten für Ihren Cluster, ist aber die langsamste der drei Methoden.

Die grundlegende Vorgehensweise bei der Online-Migration sieht wie folgt aus:

1. Ihre Anwendung verwendet die Quell-DB normal.
2. Wenn Sie zu einem Elastic Cluster migrieren, erstellen Sie Ihre Sharded-Sammlungen mithilfe des Befehls. `sh.shardCollection()` Wenn Sie zu einem instanzbasierten Cluster migrieren, fahren Sie mit dem nächsten Schritt fort.

3. Erstellen Sie Indizes vorab im Amazon DocumentDB-Cluster.
4. Erstellen Sie eine AWS DMS Aufgabe, um eine vollständige Ladung durchzuführen, und aktivieren Sie dann CDC von der MongoDB-Quellbereitstellung zum Amazon DocumentDB-Cluster.
5. Nachdem die AWS DMS Aufgabe vollständig geladen wurde und Änderungen in Amazon DocumentDB repliziert wurden, wechseln Sie vom Endpunkt der Anwendung zum Amazon DocumentDB-Cluster.



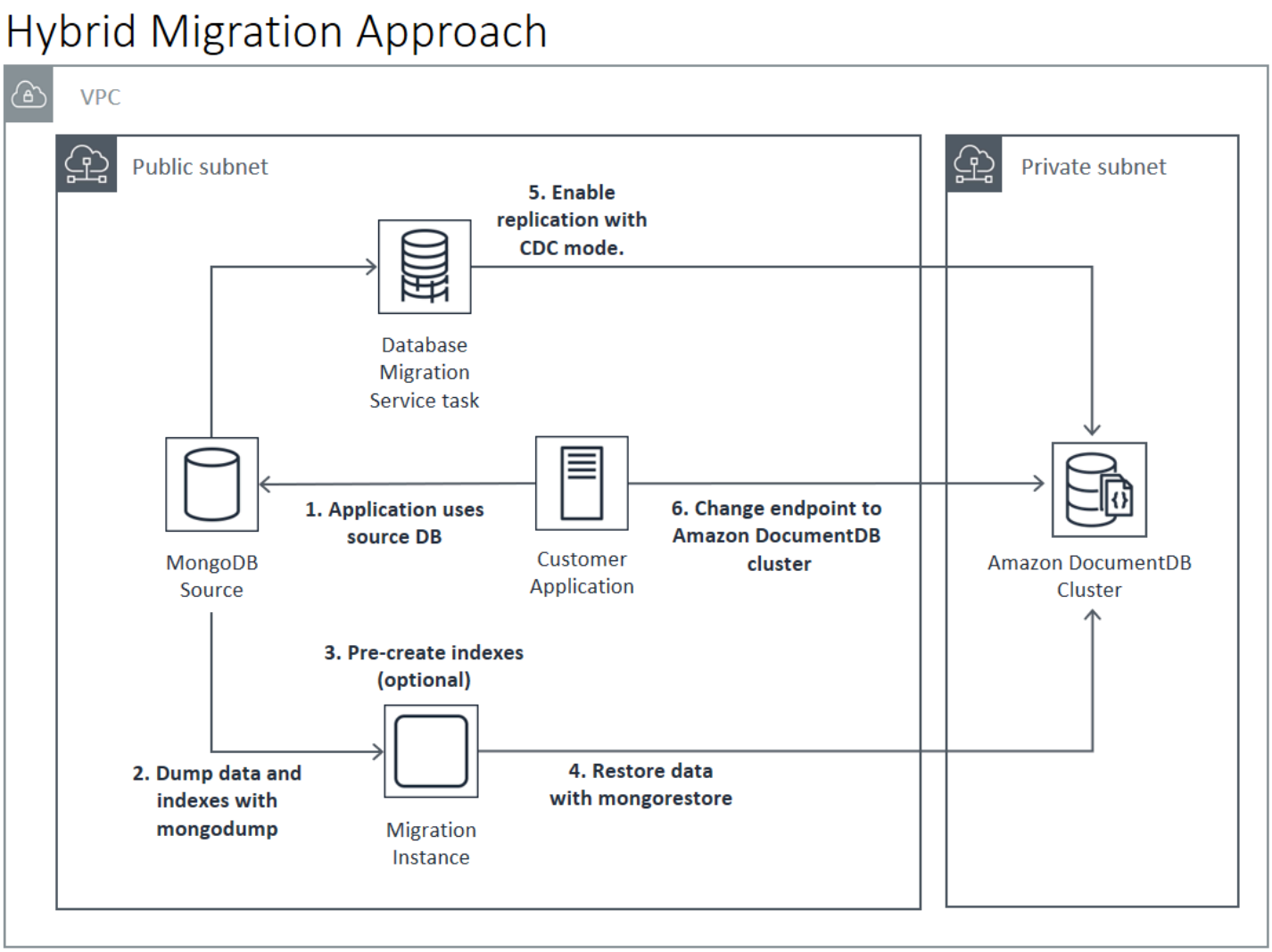
Weitere Informationen zur Verwendung AWS DMS zur Migration finden Sie unter Amazon DocumentDB als Ziel verwenden für AWS Database Migration Service und im zugehörigen Tutorial im AWS Database Migration Service Benutzerhandbuch.

Hybrid

Der hybride Ansatz verwendet die `mongorestore` Tools `mongodump` und, um Ihre Daten von Ihrer MongoDB-Quellbereitstellung zu Ihrem Amazon DocumentDB-Cluster zu migrieren. Es wird dann AWS DMS im CDC-Modus verwendet, um Änderungen zu replizieren. Der Hybrid-Ansatz ist ein Kompromiss zwischen Migrationsgeschwindigkeit und Ausfallzeit, ist jedoch auch der komplexeste der drei Ansätze.

Die grundlegende Vorgehensweise bei der Hybrid-Migration sieht wie folgt aus:

1. Ihre Anwendung verwendet die MongoDB-Quellenbereitstellung normal.
2. Dump von Sammlungsdaten und Indizes aus der MongoDB-Quellenbereitstellung
3. Stellen Sie Indizes im Amazon DocumentDB-Cluster wieder her.
4. Wenn Sie zu einem Elastic Cluster migrieren, erstellen Sie Ihre Sharded-Sammlungen mithilfe des Befehls. `sh.shardCollection()` Wenn Sie zu einem instanzbasierten Cluster migrieren, fahren Sie mit dem nächsten Schritt fort.
5. Stellen Sie die Sammlungsdaten im Amazon DocumentDB-Cluster wieder her.
6. Erstellen Sie eine AWS DMS Aufgabe, um CDC von der MongoDB-Quellbereitstellung zum Amazon DocumentDB-Cluster zu aktivieren.
7. Wenn die AWS DMS Aufgabe Änderungen innerhalb eines akzeptablen Zeitfensters repliziert, ändern Sie Ihren Anwendungsendpunkt so, dass er in den Amazon DocumentDB-Cluster schreibt.



Unabhängig vom gewählten Migrationsansatz ist es am effizientesten, vor der Migration Ihrer Daten Indizes in Ihrem Amazon DocumentDB-Cluster vorab zu erstellen. Das liegt daran, dass Amazon DocumentDB DocumentDB-Indizes Daten parallel einfügen, das Erstellen eines Indexes für bestehende Daten jedoch ein Single-Thread-Vorgang ist.

Da AWS DMS keine Indizes (nur Ihre Daten) migriert werden, ist kein zusätzlicher Schritt erforderlich, um zu vermeiden, dass Indizes ein zweites Mal erstellt werden.

Quellen für die Migration

Wenn Ihre MongoDB-Quelle ein eigenständiger Mongo-Prozess ist und Sie die Online- oder Hybridmigration verwenden möchten, konvertieren Sie zunächst Ihren eigenständigen Mongo in einen Replikatsatz, sodass der Oplog für die Verwendung als CDC-Quelle erstellt wird.

Wenn Sie von einem MongoDB-Replikatsatz oder Sharded-Cluster migrieren, sollten Sie in Betracht ziehen, für jeden als Migrationsquelle verwendete Replikatsatz oder Shard einen verketteten oder versteckten Sekundärsatz zu erstellen. Die Durchführung von Daten-Dumps kann die Auslagerung von Daten aus dem Speicher bewirken und die Leistung von Produktions-Instances beeinträchtigen. Sie können dieses Risiko reduzieren, indem Sie von einem Knoten migrieren, der keine Produktionsdaten bereitstellt.

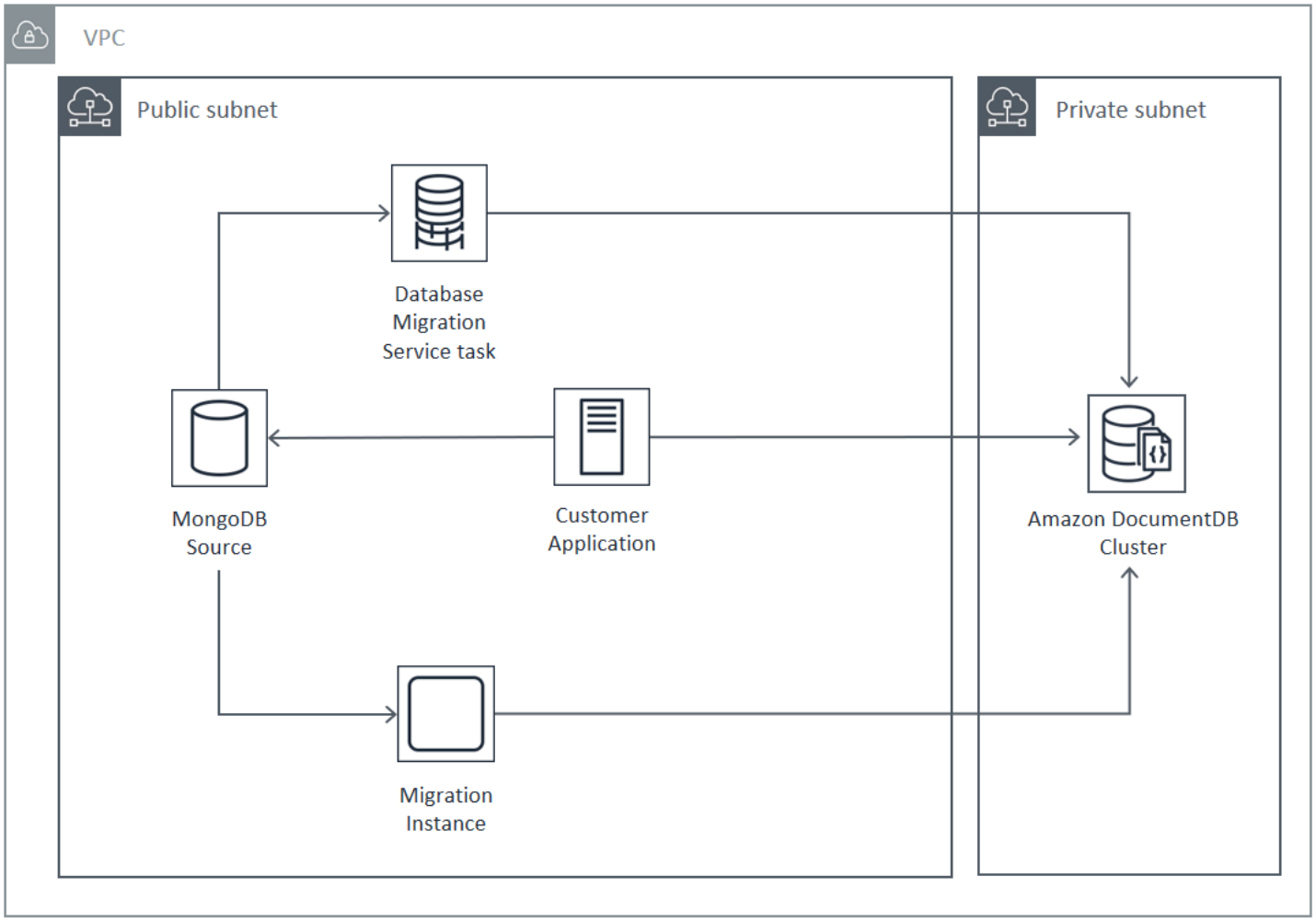
Version der Migrationsquelle

Wenn sich Ihre MongoDB-Quelldatenbankversion von der Kompatibilitätsversion Ihres Amazon DocumentDB DocumentDB-Ziel-Clusters unterscheidet, müssen Sie möglicherweise weitere Vorbereitungsschritte unternehmen, um eine erfolgreiche Migration sicherzustellen. Die beiden häufigsten Anforderungen sind die Notwendigkeit, die MongoDB-Quellinstallation auf eine unterstützte Version für die Migration (MongoDB-Version 3.0 oder höher) zu aktualisieren und Ihre Anwendungstreiber zu aktualisieren, um die Amazon DocumentDB DocumentDB-Zielversion zu unterstützen.

Wenn eine dieser Anforderungen vorliegt, sollten Sie diese Schritte zum Upgraden und Testen der Treiber in Ihren Migrationsplan aufnehmen.

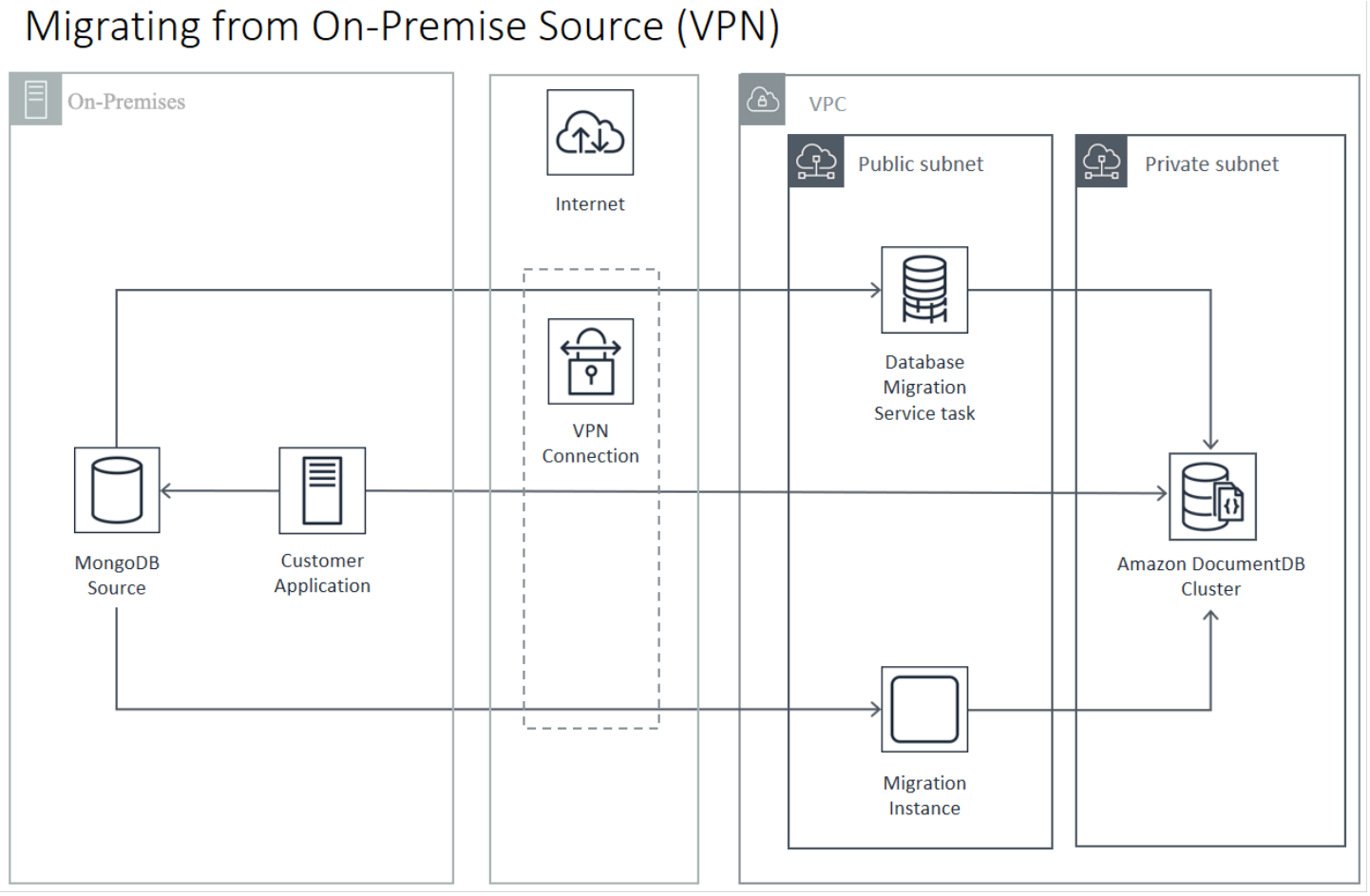
Konnektivität bei der Migration

Sie können von einer MongoDB-Quellbereitstellung, die in Ihrem Rechenzentrum ausgeführt wird, oder von einer MongoDB-Bereitstellung, die auf einer Amazon-Instance ausgeführt wird, zu Amazon DocumentDB migrieren. EC2 Die Migration von MongoDB, auf der ausgeführt wird, EC2 ist unkompliziert und erfordert lediglich, dass Sie Ihre Sicherheitsgruppen und Subnetze korrekt konfigurieren.

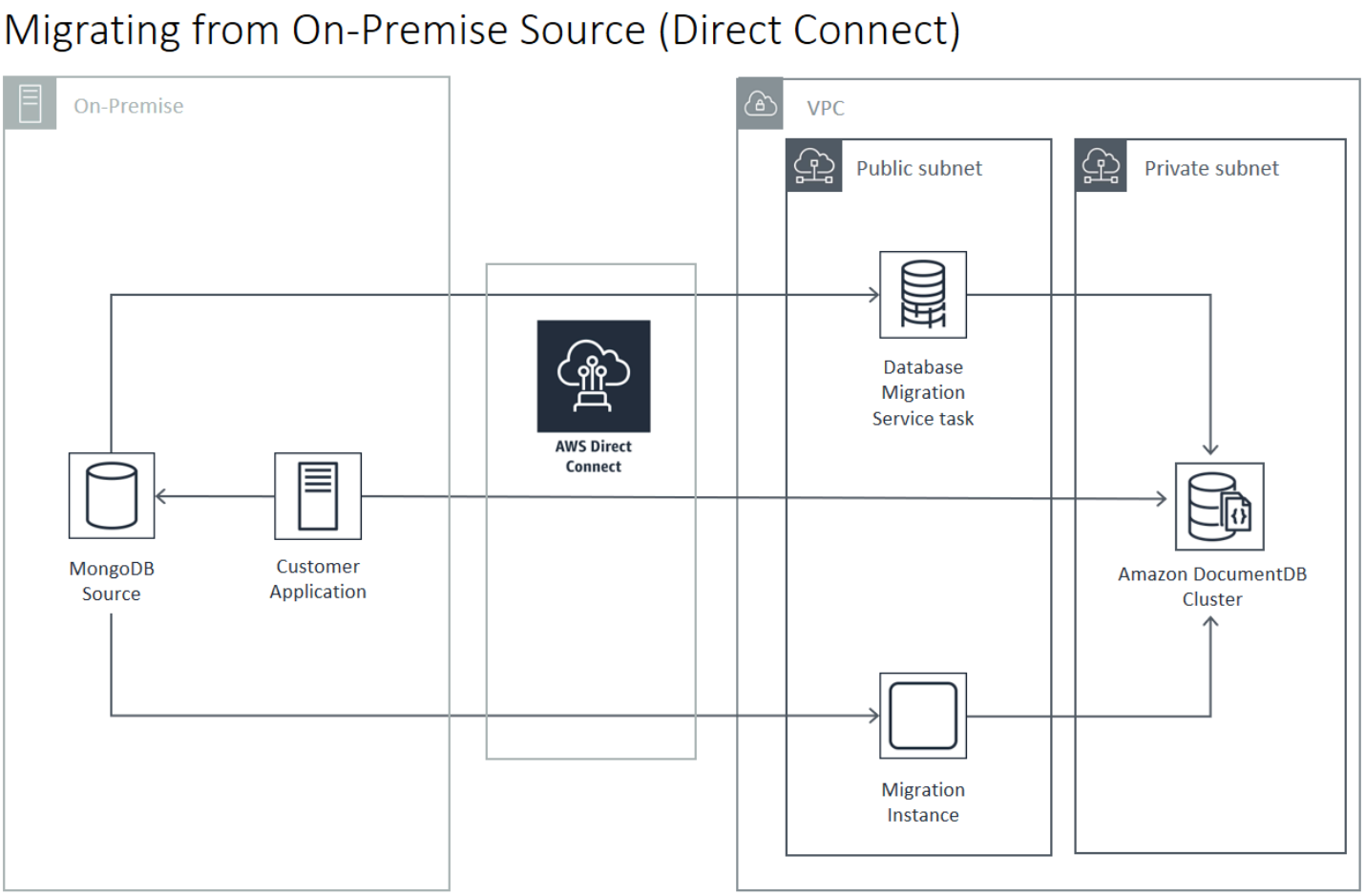


Die Migration aus einer Datenbank vor Ort erfordert eine Verbindung zwischen Ihrer MongoDB-Bereitstellung und Ihrer Virtual Private Cloud (VPC). Sie können dies über eine VPN-Verbindung (Virtual Private Network) oder mithilfe des AWS Direct Connect Dienstes erreichen. Obwohl Sie über das Internet zu Ihren VPC migrieren können, ist diese Verbindungsmethode aus Sicherheitsgründen am wenigsten wünschenswert.

Das folgende Diagramm zeigt eine Migration zu Amazon DocumentDB von einer lokalen Quelle über eine VPN-Verbindung.



Im Folgenden wird eine Migration zu Amazon DocumentDB von einer lokalen Quelle unter Verwendung von. AWS Direct Connect



Online- und Hybrid-Migrationsansätze erfordern die Verwendung einer AWS DMS Instance, die auf Amazon EC2 in einer Amazon VPC ausgeführt werden muss. Alle Ansätze erfordern einen Migrationsserver für die Ausführung von `mongodump` und `mongorestore`. Im Allgemeinen ist es einfacher, den Migrationsserver auf einer EC2 Amazon-Instance in der VPC auszuführen, in der Ihr Amazon DocumentDB-Cluster gestartet wird, da dadurch die Konnektivität zu Ihrem Amazon DocumentDB-Cluster erheblich vereinfacht wird.

Testen

Nachfolgend sind die Ziele des Pre-Migration-Tests aufgeführt:

- Überprüfen Sie, ob Ihr gewählter Ansatz das gewünschte Migrationsergebnis erzielt.
- Vergewissern Sie sich, dass der ausgewählte Instance-Typ und die Leseeinstellungsoptionen Ihren Anforderungen an die Anwendungsleistung entsprechen.
- Überprüfen Sie das Verhalten Ihrer Anwendung während des Failovers.

Überlegungen zum Testen des Migrationsplans

Beachten Sie beim Testen Ihres Amazon DocumentDB-Migrationsplans Folgendes.

Themen

- Indizes wiederherstellen
- Daten werden gelöscht
- Wiederherstellen von Daten
- Oplog-Größe
- AWS Database Migration Service Konfiguration
- Migration von einem Sharded-Cluster

Indizes wiederherstellen

`mongorestore` erstellt standardmäßig Indizes für abgerufene Sammlungen. Dies geschieht jedoch erst nach der Wiederherstellung der Daten. Insgesamt ist es schneller, Indizes in Amazon DocumentDB zu erstellen, bevor Daten im Cluster wiederhergestellt werden. Der Grund dafür ist, dass die Indizierungsoperationen während des Ladens der Daten parallelisiert werden.

Wenn Sie sich für die Vorerrichtung Ihrer Indizes entscheiden, können Sie den Schritt der Indexerstellung bei der Wiederherstellung von Daten mit `mongorestore` überspringen, indem Sie die Option `--noIndexRestore` angeben.

Daten werden gelöscht

Das `mongodump`-Tool ist die bevorzugte Methode, um Daten aus Ihrer MongoDB-Bereitstellung abzurufen. Abhängig von den Ressourcen, die für Ihre Migrations-Instance zur Verfügung stehen, können Sie `mongodump` beschleunigen, indem Sie mit der Option `--numParallelCollections` die Anzahl der parallelen Verbindungen (Standardwert: 4) erhöhen.

Wiederherstellen von Daten

Das `mongorestore` Tool ist die bevorzugte Methode, um gedumpte Daten auf Ihrer Amazon DocumentDB DocumentDB-Instance wiederherzustellen. Sie können die Wiederherstellungsleistung verbessern, indem Sie die Anzahl der Worker für jede Sammlung während der Wiederherstellung mit der Option `--numInsertionWorkersPerCollection` erhöhen. Ein Worker pro vCPU auf Ihrer primären Amazon DocumentDB-Cluster-Instance ist ein guter Anfang.

Amazon DocumentDB unterstützt die `--oplogReplay` Option des `mongorestore` Tools derzeit nicht.

Standardmäßig überspringt `mongorestore` Einfügefehler und setzt den Wiederherstellungsprozess fort. Dies kann passieren, wenn Sie nicht unterstützte Daten auf Ihrer Amazon DocumentDB DocumentDB-Instance wiederherstellen. Es kann beispielsweise vorkommen, dass Sie ein Dokument haben, das Schlüssel oder Werte mit Null-Zeichenfolgen enthält. Wenn Sie es vorziehen, dass die Operation `mongorestore` vollständig fehlschlägt, wenn ein Wiederherstellungsfehler auftritt, verwenden Sie die Option `--stopOnError`.

Oplog-Größe

Das MongoDB-Operationsprotokoll (`oplog`) ist eine gedeckelte Sammlung, die alle Datenänderungen an Ihrer Datenbank enthält. Sie können die Größe des Oplogs und den darin enthaltenen Zeitraum anzeigen, indem Sie die Operation `db.printReplicationInfo()` für einen Replikatsatz oder ein Shard-Mitglied ausführen.

Wenn Sie die Online- oder Hybridansätze verwenden, stellen Sie sicher, dass das Oplog auf jedem Replikatsatz oder Shard groß genug ist, um alle Änderungen zu enthalten, die während der gesamten Dauer des Datenmigrationsprozesses vorgenommen wurden (unabhängig davon, ob es sich um eine Volllast `mongodump` oder eine AWS DMS Aufgabe handelt), sowie einen angemessenen Puffer. Weitere Informationen finden Sie unter Überprüfen der Oplog-Größe in der MongoDB-Dokumentation. Bestimmen Sie die minimal erforderliche Oplog-Größe, indem Sie die verstrichene Zeit für den ersten Testlauf Ihres `mongodump`- oder `mongorestore`-Prozesses oder Ihrer AWS DMS -Aufgabe für das vollständige Laden erfassen.

AWS Database Migration Service Konfiguration

Das AWS Database Migration Service Benutzerhandbuch behandelt die Komponenten und Schritte, die für die Migration Ihrer MongoDB-Quelldaten auf Ihren Amazon DocumentDB-Cluster erforderlich sind. Im Folgenden finden Sie den grundlegenden Prozess für die Durchführung einer Online- oder Hybridmigration: AWS DMS

Um eine Migration durchzuführen mit AWS DMS

1. Erstellen Sie einen MongoDB-Quellendpunkt. Weitere Informationen finden Sie unter Verwendung von MongoDB als Quelle für AWS DMS.
2. Erstellen Sie einen Amazon DocumentDB DocumentDB-Zielendpunkt. Weitere Informationen finden Sie unter Arbeiten mit AWS DMS -Endpunkten.

Wenn Sie Ihren Zielendpunkt als Elastic Cluster konfigurieren, beachten Sie, dass Ihr vorhandenes Amazon DocumentDB DocumentDB-SSL-Zertifikat nicht mit elastischen Clustern funktioniert und Sie mit den folgenden Schritten ein neues SSL-Zertifikat an Ihren Endpunkt anhängen müssen:

a. Besuchen Sie https://www.amazontrust.com/repository/SFSRootCAG2.pem und speichern Sie den Inhalt als „SFSRootCAG2.pem“ -Datei. Dies ist die Zertifikatsdatei, die Sie in den nachfolgenden Schritten importieren müssen.

b. Wählen Sie bei der Erstellung des Elastic Cluster-Endpunkts unter Endpunktkonfiguration die Option Neues CA-Zertifikat hinzufügen aus.

- Geben Sie für Zertifikat-ID `SFSRootCAG2.pem` ein.
- Wählen Sie unter Import certificate file (Zertifikatsdatei importieren) die Option Choose file (Datei auswählen) und navigieren Sie zu der `SFSRootCAG2.pem`-Datei, die Sie zuvor heruntergeladen haben. Wählen Sie die Datei aus und öffnen Sie sie. Wählen Sie Zertifikat importieren und wählen Sie dann `SFSRootCAG2.pem` aus dem Drop-down-Menü Zertifikat auswählen aus.

3. Erstellen Sie mindestens eine AWS DMS Replikationsinstanz. Weitere Informationen finden Sie unter Arbeiten mit einer AWS DMS Replikationsinstanz.
4. Erstellen Sie mindestens eine AWS DMS Replikationsaufgabe. Weitere Informationen finden Sie unter Arbeiten mit AWS DMS -Aufgaben.

   Für eine Onlinemigration verwendet Ihre Migrationsaufgabe den Migrationstyp Migrate existing data and replicate ongoing changes (Migrieren von bestehenden Daten und Replizieren nachfolgender Änderungen).

   Für eine Hybridmigration verwendet Ihre Migrationsaufgabe den Migrationstyp Replicate data changes only (Nur Datenänderungen replizieren). Sie können die CDC-Startzeit so wählen, dass sie mit dem Dump-Zeitpunkt aus Ihrer `mongodump`-Operation übereinstimmt. Das MongoDB-Oplog ist idempotent. Um fehlende Änderungen zu vermeiden, ist es ratsam, einige Minuten Überlappung zwischen Ihrer `mongodump`-Endzeit und Ihrer CDC-Startzeit zu lassen.

Migration von einem Sharded-Cluster

Der Prozess für die Migration von Daten von einem MongoDB-Sharded-Cluster zu Ihrer Amazon DocumentDB DocumentDB-Instance ist im Wesentlichen der Vorgang mehrerer parallel

Replikatsatzmigrationen. Ein wichtiger Aspekt beim Testen einer Migration von Sharded-Clustern ist, dass einige Shards möglicherweise stärker genutzt werden als andere. Diese Situation führt zu unterschiedlichen Laufzeiten der Datenmigration. Stellen Sie sicher, dass Sie bei der Planung und beim Testen die Anforderungen der einzelnen Shards bewerten. `oplog`

Im Folgenden sind einige Konfigurationsprobleme aufgeführt, die bei der Migration eines Sharded-Clusters zu berücksichtigen sind:

- Bevor Sie `mongodump` ausführen oder eine AWS DMS -Migrationsaufgabe starten, müssen Sie den Sharded-Cluster-Balancer deaktivieren und warten, bis alle laufenden Migrationen abgeschlossen sind. Weitere Informationen finden Sie unter Balancer deaktivieren in der MongoDB-Dokumentation.
- Wenn Sie Daten AWS DMS replizieren, führen Sie den `cleanupOrphaned` Befehl auf jedem Shard aus, bevor Sie die Migrationsaufgaben ausführen. Wenn Sie diesen Befehl nicht ausführen, schlagen die Aufgaben möglicherweise fehl, weil das Dokument doppelt vorhanden ist. IDs Beachten Sie, dass dieser Befehl die Leistung beeinträchtigen kann. Weitere Informationen finden Sie unter cleanupOrphaned in der MongoDB-Dokumentation.
- Wenn Sie das `mongodump`-Tool zum Abrufen von Daten verwenden, sollten Sie einen `mongodump`-Prozess pro Shard ausführen. Der zeiteffizienteste Ansatz erfordert möglicherweise mehrere Migrationsserver, um die Leistung Ihres Dumps zu maximieren.
- Wenn Sie Daten AWS Database Migration Service replizieren, müssen Sie für jeden Shard einen Quellendpunkt erstellen. Führen Sie außerdem mindestens eine Migrationsaufgabe für jeden Shard aus, den Sie migrieren. Der zeiteffizienteste Ansatz erfordert möglicherweise mehrere Replikations-Instances, um die Migrationsleistung zu maximieren.

Leistungstests

Nachdem Sie Ihre Daten erfolgreich in Ihren Amazon DocumentDB-Test-Cluster migriert haben, führen Sie Ihren Test-Workload für den Cluster aus. Stellen Sie anhand von CloudWatch Amazon-Metriken sicher, dass Ihre Leistung dem aktuellen Durchsatz Ihrer MongoDB-Quellbereitstellung entspricht oder diesen übersteigt.

Überprüfen Sie die folgenden wichtigen Amazon DocumentDB-Metriken:

- Netzwerkdurchsatz
- Schreibdurchsatz
- Lesedurchsatz

- Replikatverzögerung

Weitere Informationen finden Sie unter Überwachung von Amazon DocumentDB.

Failover-Tests

Stellen Sie sicher, dass das Verhalten Ihrer Anwendung während eines Amazon DocumentDB-Failover-Ereignisses Ihren Verfügbarkeitsanforderungen entspricht. Um ein manuelles Failover eines Amazon DocumentDB-Clusters auf der Konsole zu initiieren, wählen Sie auf der Cluster-Seite im Menü Aktionen die Aktion Failover aus.

Sie können einen Failover auch einleiten, indem Sie die Operation `failover-db-cluster` von der AWS CLI aus ausführen. Weitere Informationen finden Sie `failover-db-cluster`im Abschnitt Amazon DocumentDB der AWS CLI Referenz.

Weitere Ressourcen

Weitere Informationen finden Sie in den folgenden Themen im AWS Database Migration Service - Benutzerhandbuch:

- Verwenden von Amazon DocumentDB als Ziel für AWS Database Migration Service
- Anleitung: Migration von MongoDB nach Amazon DocumentDB

Leitfaden für die Migration: MongoDB zu Amazon DocumentDB

Dieses Migrationsplaybook bietet Ihnen Ressourcen und Schritte, die Sie bei der Migration von einer MongoDB-Datenbank zu Amazon DocumentDB unterstützen.

Migrationsprozess

Im Folgenden sind die wichtigsten Schritte aufgeführt, die normalerweise bei der Migration Ihrer Daten von einer MongoDB-Datenbank zu Amazon DocumentDB erforderlich sind.

Themen

- Schritt 1: Kompatibilität und funktionale Unterschiede
- Schritt 2: Machbarkeitsnachweis

- Schritt 3: Migrieren Sie die Daten
- Schritt 4: Datenvalidierung
- Schritt 5: Umstellung der Anwendung

Schritt 1: Kompatibilität und funktionale Unterschiede

Amazon DocumentDB interagiert mit der Apache 2.0-Open-Source-Version MongoDB 3.6, 4.0 und 5.0. APIs Daher können Sie dieselben MongoDB-Treiber, -Anwendungen und -Tools mit Amazon DocumentDB verwenden, wobei nur geringe oder keine Änderungen erforderlich sind.

Der erste Schritt besteht darin, die Kompatibilität zwischen den Operatoren und Indizes, die Ihre Anwendung in Ihrer MongoDB-Datenbank verwendet, und deren Verfügbarkeit in Amazon DocumentDB zu überprüfen und die funktionalen Unterschiede zwischen ihnen zu verstehen.

Kompatibilität der Betreiber

Verwenden Sie das Amazon DocumentDB DocumentDB-Kompatibilitätstool *, um auf einfache Weise herauszufinden, ob Ihre Anwendung Operatoren verwendet, die in ihren Abfragen nicht unterstützt werden. Dieses Tool kann Ihre MongoDB-Datenbankserver-Protokolldateien oder Ihren Anwendungsquellcode scannen, um einen Bericht über nicht unterstützte Operatoren bereitzustellen. Wenn Sie feststellen, dass nicht unterstützte Operatoren verwendet werden, müssen Sie Ihre Anwendung ändern, um nicht unterstützte Operatoren zu umgehen.

Um die Kompatibilität zwischen den in Ihrem Setup verwendeten MongoDB-Operatoren und den unterstützten Amazon DocumentDB DocumentDB-Operatoren zu überprüfen, führen Sie Folgendes aus:

```
git clone https://github.com/awslabs/amazon-documentdb-tools.git
cd amazon-documentdb-tools/compat-tool/
python3 compat.py --version <Amazon DocumentDB version> --directory <mongodb logfiles/
source code>
```

Weitere Informationen finden Sie unter Unterstützte MongoDB APIs, Operationen und Datentypen in Amazon DocumentDB.

* Nicht offiziell unterstützt von. AWS

Kompatibilität der Indizes

Sie können das Amazon DocumentDB-Indextool* verwenden, um herauszufinden, ob Sie Indextypen verwenden, die in Amazon DocumentDB nicht unterstützt werden. Dieses Tool benötigt eine Verbindung zu Ihrer Quelldatenbank, um Indexdefinitionen lesen zu können.

Dazu müssen Sie zunächst die Indexdefinitionen mithilfe der `--dump-indexes` Option in ein Verzeichnis speichern. Führen Sie dann das Tool mit der `--show-issues` Option aus und geben Sie das Verzeichnis an, in dem inkompatible Indizes gefunden werden können.

Indizes exportieren:

```
git clone https://github.com/awslabs/amazon-documentdb-tools.git
sudo pip install -r amazon-documentdb-tools/index-tool/requirements.txt
mkdir <directory to dump index definitions>
python3 migrationtools/documentdb_index_tool.py --dump-indexes --dir <directory> --uri
 <source-mongodb-uri>
```

Suchen Sie nach inkompatiblen Indizes:

```
python3 migrationtools/documentdb_index_tool.py --show-issues --dir <dumped-index-
definitions-directory>
```

Wenn Sie feststellen, dass Indextypen verwendet werden, die nicht unterstützt werden, müssen Sie Ihre Anwendung oder Ihr Datenmodell ändern, um das Problem zu umgehen oder ohne die inkompatiblen Indizes fortzufahren.

Weitere Informationen zu den unterstützten Indextypen und Eigenschaften in Amazon DocumentDB finden Sie unter Indizes und Indexeigenschaften und So indexieren Sie in Amazon DocumentDB.

* Nicht offiziell unterstützt von. AWS

Funktionelle Unterschiede

Lesen Sie Funktionale Unterschiede zu MongoDB weiter, um sich mit den Unterschieden vertraut zu machen.

Schritt 2: Machbarkeitsnachweis

Führen Sie einen Machbarkeitsnachweis durch, indem Sie Ihre Anwendung oder Ihre reguläre Testsuite auf Amazon DocumentDB ausführen, um Funktionalität und Leistung zu testen. Möglicherweise müssen Sie Ihren Amazon DocumentDB-Cluster mit Daten füllen, um die Tests

durchzuführen. Sie können beispielsweise die `mongorestore` Tools `mongodump` und verwenden, um Daten aus Ihrer MongoDB-Quelle zu kopieren.

Funktionale Tests

Erstellen Sie einen Amazon DocumentDB-Cluster (siehe Einen Amazon DocumentDB-Cluster erstellen) und führen Sie Ihre Anwendung oder Ihre Funktionstestsuite aus, um zu überprüfen, ob alle Anwendungsworkflows weiterhin reibungslos in Amazon DocumentDB funktionieren.

Leistungstests

Führen Sie Leistungstests für Ihre Anwendung oder Leistungstestsuite aus, die auf Amazon DocumentDB ausgeführt wird, mit einer Arbeitslast, die Ihrer Produktions-Workload ähnelt, um festzustellen, ob das Setup Ihre Latenzanforderungen erfüllt. Passen Sie Ihre Arbeitslast an die Leistung an oder skalieren Sie Ihren Amazon DocumentDB-Cluster nach Bedarf. Weitere Informationen erhalten Sie unter Problembehebung bei Leistung und Ressourcennutzung und Skalierung von Amazon DocumentDB-Clustern.

Es ist wichtig, dass Sie Ihren Amazon DocumentDB-Cluster mit den richtigen Instance-Typen dimensionieren, um eine optimale Leistung zu erzielen. Weitere Informationen finden Sie unter Bewährte Methoden für Dimensionierung der Instanzen.

Failover-Tests

Möglicherweise möchten Sie beobachten, wie Ihre Anwendung auf einen Neustart des Amazon DocumentDB-Primärknotens, einen primären Knoten-Failover oder eine Löschung des primären Knotens in einem Cluster mit mehreren Knoten reagiert und wenn Replikatknoten neu gestartet oder entfernt werden. Auf diese Weise können Sie überprüfen, ob Ihre Anwendung diesen Ereignissen standhält. Weitere Informationen finden Sie unter Testen eines Failovers.

Informationen zu den Ausnahmen, die eine Anwendung tolerieren sollte, und zu deren effizientem Umgang finden Sie unter Erstellen robuster Anwendungen mit Amazon DocumentDB.

> **Note**
>
> Es gibt keinen Ersatz dafür, Ihre Arbeitslast auf Amazon DocumentDB zu testen

Schritt 3: Migrieren Sie die Daten

Migrieren Sie Ihre Daten nach einem erfolgreichen Machbarkeitsnachweis zu Amazon DocumentDB. Die meisten unserer Kunden verwenden Online- oder Offline-Migrationsansätze, um ihre Daten zu migrieren.

Online-Migration

Mithilfe der Online-Migrationsmethode können Sie Daten aus Ihrer Quelldatenbank, die von einigen Gigabyte bis zu mehreren Terabyte reichen, nahezu ohne Ausfallzeiten zu Amazon DocumentDB migrieren. Weitere Informationen finden Sie unter ().AWS Database Migration ServiceAWS DMS

Wenn Sie aus einer MongoDB-Datenbank migrieren, können Sie sie verwenden, AWS DMS um eine Volllast durchzuführen und laufende Änderungen zu replizieren.

Ein step-by-step Verfahren finden Sie unter Migration zu Amazon DocumentDB mit der Online-Methode.

Weitere Informationen finden Sie im AWS Database Migration Service Abschnitt Amazon DocumentDB als Ziel verwenden im AWS Database Migration Service Benutzerhandbuch.

Zu beachtende Punkte bei AWS DMS:

- Segmentierung: Bei der Migration von Datenbanken mit mehreren Terabyte kann es mit den Standardeinstellungen langsam sein AWS DMS, da die Volllast von DMS standardmäßig per Single-Threading pro Sammlung erfolgt, was zu längeren Migrationszeiten führt. Um die Volllast bei großen Datenbankmigrationen zu beschleunigen, können Sie die Segmentierungsfunktion in verwenden. AWS DMS

  Weitere Informationen zur Verwendung der Segmentierung mit finden Sie unter Verwenden der automatischen Segmentierung mit AWS DMS. AWS DMS
- DMS-Instanztyp: Um die Datenmigration zu beschleunigen, müssen Sie die richtige DMS-Instanz auswählen.

Offline-Migration

Die Offline-Migration ist der einfachste Ansatz, um Datenbanken zu Amazon DocumentDB zu verschieben. Dieser Ansatz wird hauptsächlich für POCs und für Workloads verwendet, bei denen es während der Migration zu Schreibausfällen kommen kann.

Einen step-by-step Prozess finden Sie unter Migrieren von MongoDB zu Amazon DocumentDB mithilfe der Offline-Methode.

Schritt 4: Datenvalidierung

Sobald die Daten erfolgreich migriert wurden, überprüfen Sie die Daten auf ihre Richtigkeit, um Vertrauen zu gewinnen. In der Konsole für AWS DMS Migrationsaufgaben finden Sie die Metriken für migrierte Daten. Weitere Informationen finden Sie unter Überprüfen der migrierten Daten.

Sie können auch das Amazon DocumentDB DataDiffer Tool * verwenden, um die Datenkonsistenz zwischen den Quell- und Zielsammlungen zu überprüfen.

* Nicht offiziell unterstützt von AWS.

Schritt 5: Umstellung der Anwendung

Dazu müssen Sie die Datenbankverbindungszeichenfolge Ihrer Anwendung ändern, um Ihren Amazon DocumentDB-Cluster zu verwenden.

Weitere Informationen zum Herstellen einer Verbindung mit Amazon DocumentDB finden Sie unterAls Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen.

Online-Migration

Nachdem das vollständige Laden der Daten abgeschlossen ist, repliziert die laufenden Änderungen AWS DMS weiterhin von Ihrer Quelle nach Amazon DocumentDB. Nachdem die Änderungen erfasst wurden und Ihre Datenvalidierungsprüfungen abgeschlossen sind, können Sie eine Umstellung auf Amazon DocumentDB durchführen.

Offline-Migration

Sobald Sie die vollständigen Datenlade- und Datenvalidierungsprüfungen abgeschlossen haben, können Sie die Umstellung auf Amazon DocumentDB durchführen.

Weitere Ressourcen

Hier sind einige zusätzliche Ressourcen, die Ihnen bei der Migration helfen könnten:

- Video: Bewährte Methoden für die Migration zu Amazon DocumentDB
- Video: Erste Schritte mit Amazon DocumentDB Observability and Monitoring
- Zusätzliche Dienstprogramme: Amazon DocumentDB Tools *

- Leitfaden für Entwickler zur Migration: Migration zu Amazon DocumentDB

* Nicht offiziell unterstützt von AWS.

Direktes Upgrade der Hauptversion von Amazon DocumentDB

Amazon DocumentDB stellt neue Versionen von Datenbank-Engines erst nach umfangreichen Tests allgemein zur Verfügung. Sie können wählen, wie und wann Sie Ihre Amazon DocumentDB-Cluster auf die neue Version aktualisieren möchten.

Derzeit unterstützt Amazon DocumentDB drei Hauptversionen: Amazon DocumentDB 3.6, 4.0 und 5.0. Sie können ein direktes Major Version Upgrade (MVU) Ihrer Datenbank durchführen und dabei dieselben Endpunkte, Speicher und Tags der Cluster beibehalten und Ihre Anwendungen ohne Änderungen weiter verwenden. Diese Funktion ist in allen Regionen, in denen Amazon DocumentDB 5.0 verfügbar ist, kostenlos verfügbar.

> **Important**
>
> Ihre Amazon DocumentDB-Cluster sind während des direkten Upgrades der Hauptversion nicht verfügbar und Ihre Cluster werden mehrfach neu gestartet. Bitte stellen Sie nach dem Start des Upgrades keine Verbindung zum Cluster her, lesen oder darauf schreiben. Die Ausfallzeit des Upgrades kann je nach Anzahl der Sammlungen, Indizes, Datenbanken und Instanzen von Cluster zu Cluster variieren. Wir empfehlen, das Upgrade während Ihres Wartungsfensters oder zu Zeiten mit geringer Auslastung durchzuführen. Sobald Ihr Cluster aktualisiert wurde, können Sie den Cluster nicht mehr auf die vorherige Version herunterstufen. Sie können sich jedoch dafür entscheiden, Ihren Snapshot vor dem Upgrade auf einem neuen Cluster wiederherzustellen.

Themen

- MVU-Voraussetzungen und Einschränkungen
- Bewährte Methoden für direkte Upgrades von Hauptversionen
- Durchführung eines direkten Hauptversions-Upgrades
- Unterschiede zwischen auf Amazon DocumentDB 3.6/4.0 auf 5.0 aktualisierten Clustern und neuen Amazon DocumentDB 5.0-Clustern
- Fehlerbehebung bei einem direkten Upgrade der Hauptversion

MVU-Voraussetzungen und Einschränkungen

Im Folgenden sind die Voraussetzungen und Einschränkungen für ein direktes Upgrade einer Hauptversion aufgeführt, mit denen Sie möglicherweise vertraut sein müssen und die Sie beachten müssen, bevor Sie das Upgrade durchführen:

- Instanztyp — Amazon DocumentDB 4.0/5.0 unterstützt keine r4.*-Instances. Um mit einem direkten Upgrade der Hauptversion fortzufahren, ändern Sie r4.*-Instances in r5.*-Instances. Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB DocumentDB-Instance. Informationen zu unterstützten Instances, Unterstützte Instance-Klassen nach Region die auf der Amazon DocumentDB DocumentDB-Engine-Version basieren, finden Sie unter.
- Instance-Betriebssystem-Patches — Für ein direktes Upgrade der Hauptversion ist der neueste Betriebssystem-Patch (OS) erforderlich, um fortzufahren. Bitte wenden Sie alle ausstehenden Betriebssystem-Wartungsmaßnahmen auf die Instanzen an, bevor Sie mit dem direkten Upgrade fortfahren. Weitere Informationen finden Sie unter Betriebssystem-Updates für Amazon DocumentDB.

  > **Note**
  >
  > In manchen Situationen sind Instanz-Betriebssystem-Patches nicht sichtbar, wenn auf Cluster-Ebene Engine-Patches ausstehen. Möglicherweise müssen Sie Engine-Patches auf Clusterebene installieren, bevor Sie mit der Installation von Instanz-Betriebssystem-Patches und anschließend mit dem direkten Upgrade der Hauptversion fortfahren können. Siehe Durchführen eines Patch-Updates für die Engine-Version eines Clusters.

- Ein direktes Upgrade der Hauptversion ist in allen Regionen verfügbar, in denen Amazon DocumentDB 5.0 verfügbar ist.
- Ein direktes Upgrade der Hauptversion wird mit Amazon DocumentDB 4.0 als Zielversion nicht unterstützt.
- Ab Amazon DocumentDB 4.0 wird „.“ in Benutzernamen nicht mehr unterstützt. Wenn Sie ein Upgrade von Amazon DocumentDB 3.6 auf 5.0 durchführen und einen Benutzernamen haben, der“ . „, bitte erstelle deinen Nutzernamen neu ohne“ . „, bevor Sie mit der MVU vor Ort fortfahren.
- Ein direktes Upgrade der Hauptversion wird derzeit auf globalen Clustern und Elastic Clustern von Amazon DocumentDB nicht unterstützt.

> **Note**
>
> Um Ihre globalen Cluster zu aktualisieren, löschen Sie Ihre sekundären Cluster aus dem globalen Cluster, konvertieren Sie den primären Cluster in einen regionalen Cluster, führen Sie ein direktes Upgrade der Hauptversion auf dem regionalen (primären) Cluster durch und erstellen Sie dann den globalen Cluster neu, indem Sie sekundäre Cluster mit demselben Namen hinzufügen, um dieselben Endpunkte wie zuvor beizubehalten. Beachten Sie, dass I/O-Gebühren anfallen, während Ihr aktualisierter primärer Cluster Daten auf Ihre neu hinzugefügten sekundären Cluster repliziert. Ausführliche Schritte zum Entfernen von sekundären Clustern vor dem Löschen aus dem globalen Cluster finden Sie unter. Einen Cluster aus einem globalen Amazon DocumentDB-Cluster entfernen

- Wenn Sie über eine große Anzahl von Indizes (> 3.000) verfügen, die in Instances mit hoher Leistung betrieben werden (z. B. t3.medium oder t4g.medium), müssen Sie Ihre primäre Instance auf eine größere Instance skalieren (z. B. mindestens r5.large), um das direkte Upgrade der Hauptversion durchzuführen. Sie können die Instance-Größe nach unten skalieren, sobald Ihr direktes Upgrade der Hauptversion abgeschlossen ist. In der folgenden Tabelle finden Sie die maximale Anzahl von Indizes, die für die Instance-Typen db.t3 und db.t4g für ein direktes Upgrade der Hauptversion unterstützt werden:

Instance	Maximale Anzahl der unterstützten Indizes für In-Place-MVU
db.t4g.medium	3.000
db.t3.medium	10 K

Bewährte Methoden für direkte Upgrades von Hauptversionen

Themen

- Testen Sie direkte Hauptversions-Upgrades mit geklonten Clustern
- Vor einem direkten Upgrade der Hauptversion
- Während eines direkten Hauptversions-Upgrades
- Nach einem direkten Upgrade der Hauptversion

Testen Sie direkte Hauptversions-Upgrades mit geklonten Clustern

1. Um Hauptversionsupgrades vor Ort zu testen, empfehlen wir die Verwendung der Schnellklonfunktion, um einen Klon Ihres Zielclusters zu erstellen. Es fallen keine Speicherkosten an, wenn Sie das direkte Upgrade der Hauptversion auf einem geklonten Volume testen, es sei denn, Sie ändern Daten auf dem Cluster. Weitere Informationen zum Klonen von Volumes finden Sie unter. Klonen eines Volumes für einen Amazon DocumentDB-Cluster
2. Um eine realistischere Schätzung der Zeit zu erhalten, die für das direkte Upgrade der Hauptversion benötigt wird, ordnen Sie die Anzahl der Instanzen des geklonten Clusters dem Zielcluster zu.
3. Wir empfehlen, den neu aktualisierten Amazon DocumentDB 5.0-Cluster vollständig auf Funktionsunterschiede zu testen, um sicherzustellen, dass alles wie erwartet funktioniert.

Vor einem direkten Upgrade der Hauptversion

1. Halten Sie eine versionskompatible Cluster-Parametergruppe bereit.

   Verwenden Sie die standardmäßige Cluster-Parametergruppe von Amazon DocumentDB für die neue Engine-Version oder erstellen Sie Ihre eigene benutzerdefinierte Cluster-Parametergruppe für die neue Engine-Version.

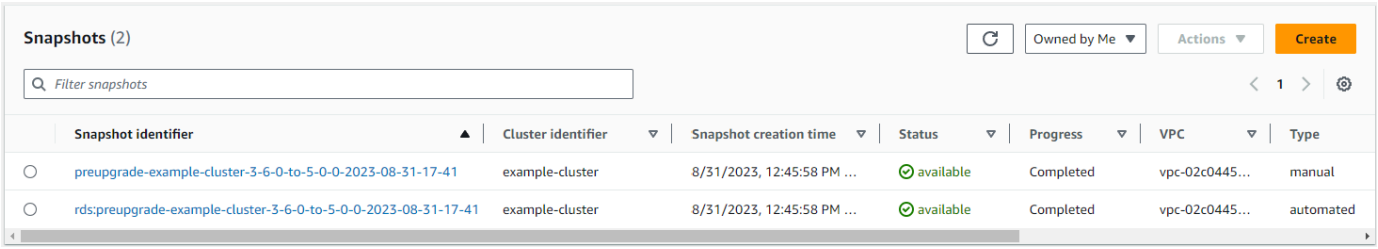
   Wenn Sie im Rahmen der Upgrade-Anfrage eine Amazon DocumentDB-Cluster-Parametergruppe zuordnen, wird der Cluster beim direkten Upgrade der Hauptversion automatisch neu gestartet, um die neue Parametergruppe anzuwenden.
2. Stellen Sie sicher, dass Sie die Voraussetzungen für ein direktes Upgrade der Hauptversion erfüllen, wie im Abschnitt Voraussetzungen und Einschränkungen beschrieben.
3. Erstellen Sie einen manuellen Snapshot.

   Der Upgrade-Prozess erstellt während des Upgrades einen Snapshot Ihres Datenbank-Clusters. Es wird dringend empfohlen, vor dem Upgrade-Prozess einen eigenen manuellen Snapshot zu erstellen. Siehe Einen manuellen Cluster-Snapshot erstellen.

   > **Note**
   >
   > Der durch den Upgrade-Prozess erstellte auto Snapshot wird nicht automatisch gelöscht, nachdem das direkte Upgrade der Hauptversion abgeschlossen ist. Für diesen Snapshot fallen keine Gebühren an, solange er sich innerhalb der Aufbewahrungsfrist befindet. Sie

können sich dafür entscheiden, diesen Snapshot zu löschen, sobald Sie bestätigt haben, dass Ihr Cluster erfolgreich aktualisiert wurde.

Der Snapshot ist benannt als`preupgrade-<name>-<version>-<timestamp>`.



4. Überprüfen Sie, ob Sie bereits ein direktes Upgrade der Hauptversion Ihres Clusters geplant haben.

   Wenn Sie den Cluster geändert und ausgewählt haben, ihn im nächsten Wartungsfenster anzuwenden, ist der Zeitplan für das direkte Upgrade der Hauptversion nicht auf der Konsole sichtbar, Sie können ihn jedoch in der CLI anzeigen. Sie können den [describe-db-clusters]Befehl ausführen, um zu überprüfen, ob ein direktes Upgrade der Hauptversion bereits geplant ist:

   ```
   aws docdb describe-db-cluster \
     --region us-east-1 \
     --db-cluster-identifier mydocdbcluster
   ```

   Ersetzen Sie im obigen Beispiel jedes einzelne *user input placeholder* durch die Informationen Ihres Clusters.

   Der Befehl gibt die folgende Ausgabe zurück:

   ```
   "PendingModifiedValues": {
       "EngineVersion": "5.0.0"
   },
   ```

5. Führen Sie mehrere Testläufe mithilfe von Volume Clone in niedrigeren Umgebungen durch, um den Cluster nach dem direkten Upgrade der Hauptversion auf etwaige Ausführungspläne und funktionale Unterschiede zu testen. Wir empfehlen das Klonen mit derselben Anzahl und Größe von Instanzen, um eine bessere Schätzung der Laufzeit des direkten Upgrades der Hauptversion

zu erhalten. Weitere Informationen finden Sie unter Klonen eines Volumes für einen Amazon DocumentDB-Cluster.

6. Wenn der vorherige Schritt erfolgreich war, fahren Sie mit dem direkten Upgrade der Hauptversion auf dem Produktionscluster fort.

Während eines direkten Hauptversions-Upgrades

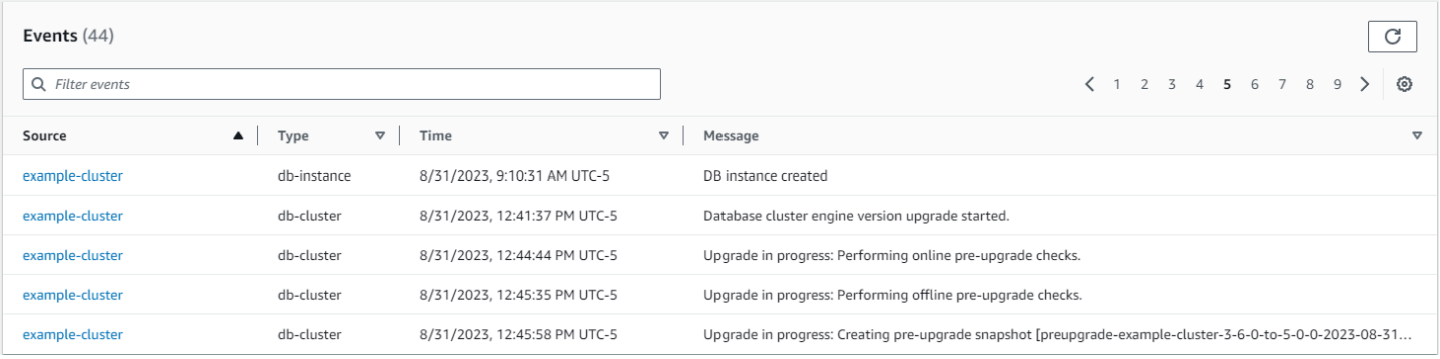
Sie können den Fortschritt Ihres direkten Hauptversions-Upgrades überwachen, indem Sie Cluster-Wartungsereignisse abonnieren. Wenn das Upgrade abgeschlossen ist, erhalten Sie die Meldung „Die Datenbankcluster-Hauptversion wurde aktualisiert“. Dieses und andere Ereignisse, die während des Upgrades auftreten, werden im Abschnitt „Ereignisse und Tags“ der Cluster-Detailseite in der Amazon DocumentDB DocumentDB-Konsole angezeigt. Der Cluster-Status ändert sich dann von „Upgrade“ zu „Verfügbar“.

Über die CLI können Sie ausführen, `aws docdb create-event-subscription` um Ereignisse zu erstellen und `aws docdb describe-events` den Fortschritt zu überwachen. Sie können auch Ereignisbenachrichtigungen für die oben genannten Ereignisse an Amazon SNS als Ziel einrichten, das per E-Mail, Push-Nachrichten und anderen Methoden benachrichtigt werden soll. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Veranstaltungen abonnieren.

Ein direktes Upgrade der Hauptversion generiert während des Upgrades die folgenden Ereignisse:

- <cluster-name><timestamp>Aktualisierung läuft: Snapshot vor dem Upgrade wird erstellt [Preupgrade- -]
- Aktualisierung läuft: Volume wird geklont.
- Aktualisierung läuft: Writer wird aktualisiert.
- Aktualisierung läuft: Lesegeräte werden aktualisiert.
- Die Hauptversion des Datenbankclusters wurde aktualisiert.

Ereignisse sind auch auf der Konsole unter der Seite Ereignisse sichtbar:



In der können Sie den `describe-events`Befehl ausführen AWS CLI, um den Fortschritt zu verfolgen:

```
aws docdb describe-events
  --source-identifier mydocdbcluster
  --source-type db-cluster
```

Ersetzen Sie im obigen Beispiel jeden *`user input placeholder`* durch die Informationen Ihres Clusters.

Der Befehl gibt die folgende Ausgabe zurück:

```
{
    "Events": [
        {
            "SourceIdentifier": "mydocdbcluster",
            "SourceType": "db-cluster",
            "Message": "Database cluster engine version upgrade started.",
            "EventCategories": [
                "maintenance"
            ],
            "Date": "2023-07-11T23:20:32.444000+00:00",
            "SourceArn": "arn:aws:rds:us-east-1:xxxx:cluster:mycluster"
        }
    ]
}
```

Nach einem direkten Upgrade der Hauptversion

Fügen Sie für Amazon DocumentDB 3.6 dem Cluster ein Tag hinzu, um zu unterscheiden, dass der Cluster von Amazon DocumentDB 3.6 auf Amazon DocumentDB 5.0 aktualisiert wurde und nicht

ein neu erstellter Amazon DocumentDB 5.0-Cluster. Weitere Informationen finden Sie im Abschnitt zu den Unterschieden zwischen einem aktualisierten Amazon DocumentDB 5.0-Cluster und einem neuen Amazon DocumentDB 5.0-Cluster.

Erstellen Sie nach Abschluss der Installation der MVU einen manuellen Snapshot für den Fall, dass Sie den Zustand nach dem Upgrade wiederherstellen müssen. Der automatische Snapshot-Vorgang wird fortgesetzt, sobald das direkte Upgrade der Hauptversion abgeschlossen ist. Für den manuellen Snapshot fallen keine Gebühren an, solange der Aufbewahrungszeitraum eingehalten wird.

Um die neuen Funktionen von Amazon DocumentDB 5.0 nutzen zu können, z. B. die clientseitige Verschlüsselung auf Feldebene, empfehlen wir, Ihre Treiberversion auf die MongoDB 5.0 API-Version zu aktualisieren. Weitere Informationen finden Sie unter Was ist neu in Amazon DocumentDB 5.0 eine Liste der Funktionen von Amazon DocumentDB 5.0.

> **Important**
>
> Unmittelbar nach der Durchführung eines direkten Major Version Upgrades (MVU) füllt Ihr Amazon DocumentDB 5.0-Cluster die Index-Metadaten erneut auf, auf deren Grundlage die Datenbank-Engine die Abfrageausführungspläne optimiert. Die erwartete Abfrageleistung auf Ihrem Amazon DocumentDB-Cluster wird wieder aufgenommen, nachdem die Neuberechnung der Index-Metadaten abgeschlossen ist. In der Regel ist dieser Vorgang in wenigen Minuten abgeschlossen, kann aber je nach Anzahl der Indizes in Ihrem Cluster bis zu zwei Stunden dauern. Ein sofortiger Neustart, ein Failover oder ein Hoch-/Herunterskalieren Ihrer Writer-Instanz nach der Installation der MVU können den Berechnungsprozess der Index-Metadaten auf Ihrem Cluster stören. Nach Abschluss der In-Place-MVU empfehlen wir, solche Änderungen vorzunehmen, sobald Sie die erwartete Abfrageleistung auf Ihrem Amazon DocumentDB 5.0-Cluster beobachtet haben.
> Darüber hinaus werden nach Abschluss der integrierten MVU die verfügbaren Change-Stream-Daten auf die letzten 3 Stunden begrenzt.
> Bitte wenden Sie sich an den AWS Support, wenn Sie feststellen, dass dieser vorübergehende Leistungsabfall nach der Installation der MVU länger als zwei Stunden anhält.

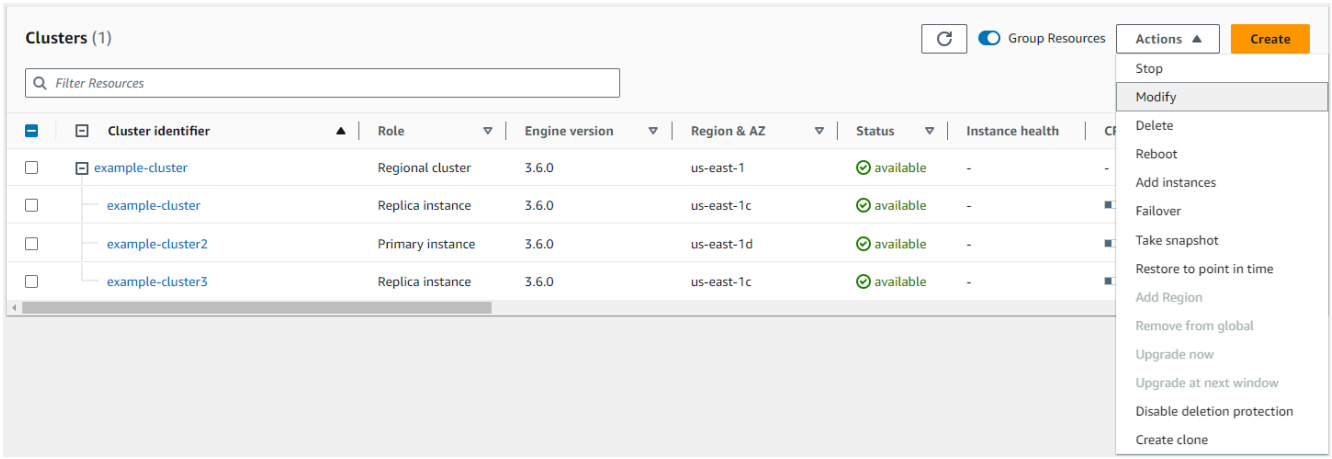
Testen Sie den aktualisierten Amazon DocumentDB 5.0-Cluster vollständig, um sicherzustellen, dass alles wie erwartet funktioniert.

Durchführung eines direkten Hauptversions-Upgrades

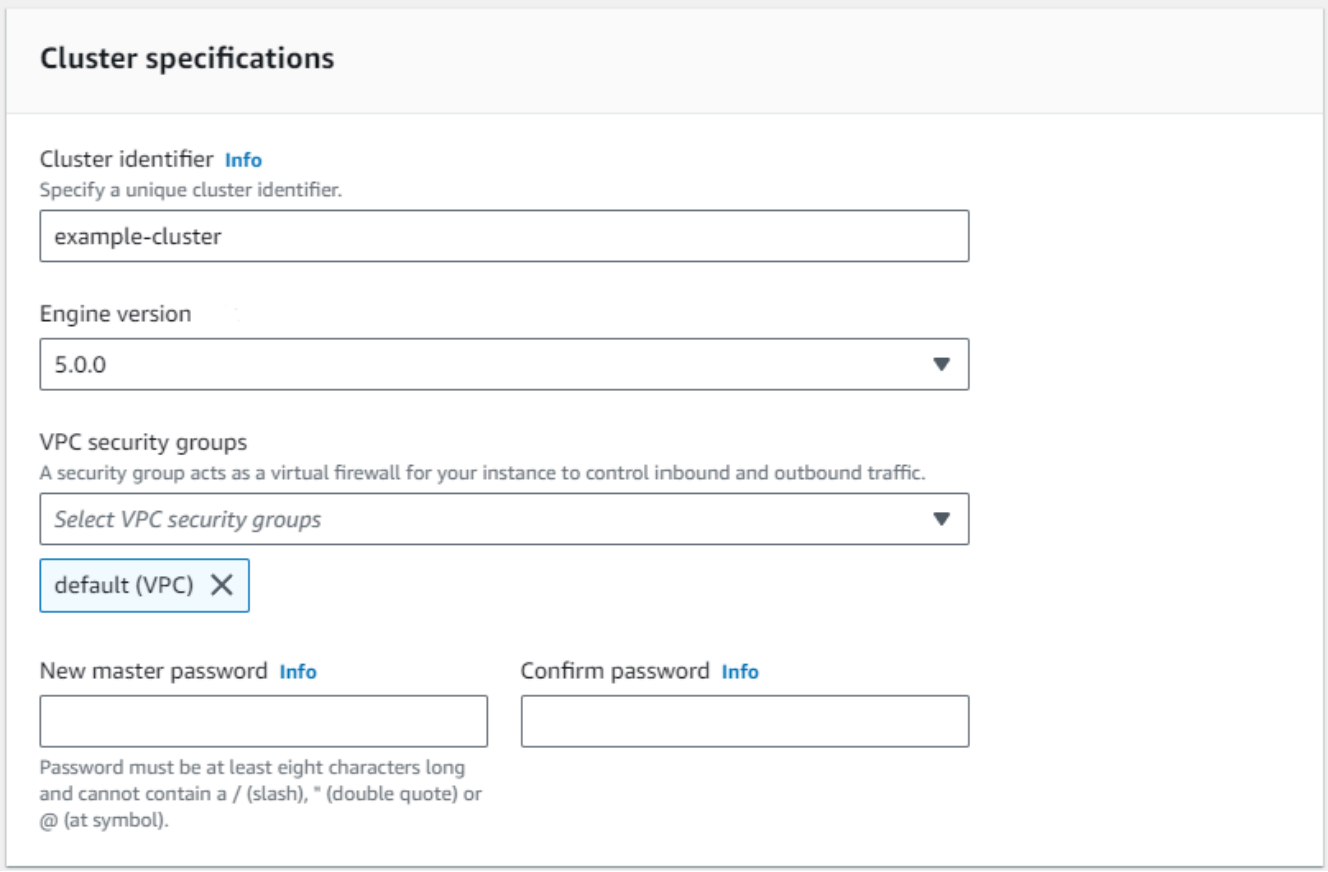
Using the AWS Management Console

Um ein direktes Upgrade der Hauptversion durchzuführen, verwenden Sie: AWS Management Console

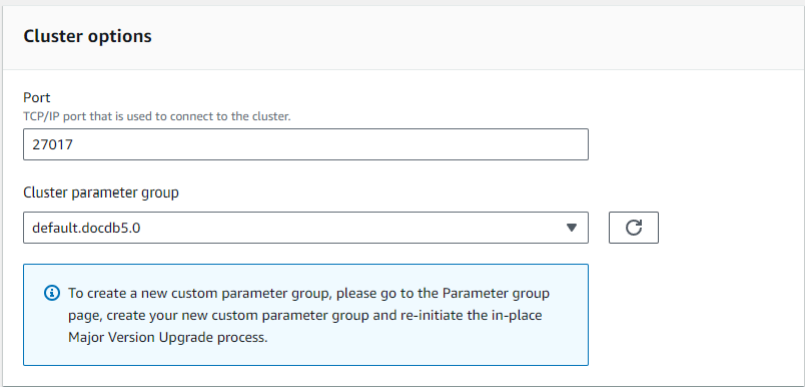
1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Wählen Sie in der Cluster-Tabelle den Quell-Cluster aus, klicken Sie auf Aktionen und dann auf Ändern.



3. Wählen Sie im Dialogfeld „Cluster ändern“ im Abschnitt „Clusterspezifikationen“ die Zieldatenbankversion (5.0.0) aus dem Dropdownmenü Engine-Version aus.

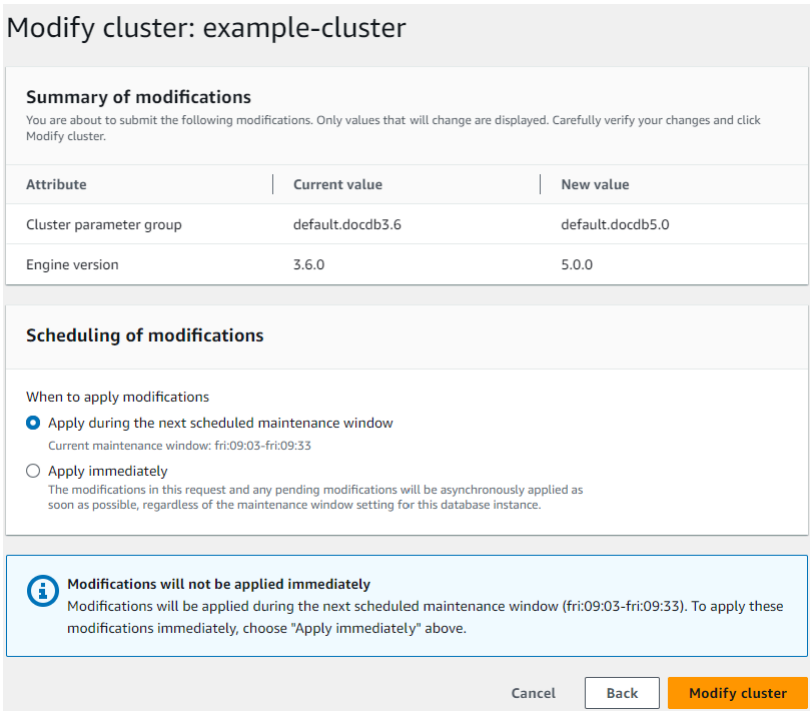


4. Wählen Sie im Abschnitt Clusteroptionen die entsprechende Cluster-Parametergruppe (default.docdb5.0) oder eine benutzerdefinierte Parametergruppe aus.

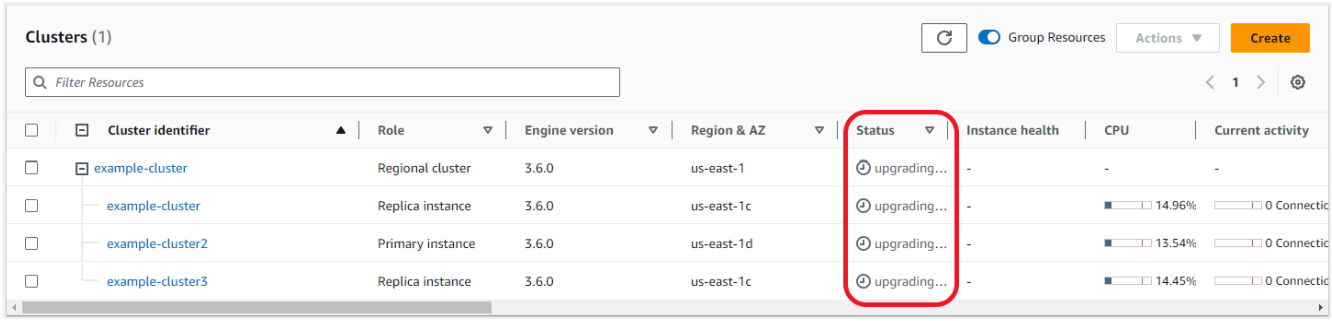


5. Wenn Sie fertig sind, scrollen Sie nach unten und wählen Sie Weiter.
6. Wählen Sie im Abschnitt Planung von Änderungen Ihren bevorzugten Terminplan aus: sofort anwenden oder im nächsten Wartungsfenster anwenden.

   Wählen Sie Cluster ändern aus.



7. Notieren Sie sich in der Clustertabelle den Status Ihres Clusters während der Aktualisierung:



Using the AWS CLI

Verwenden Sie den [modify-db-cluster](#)Befehl mit der gewünschten Engine-Versionsoption und dem `allow-major-version-upgrade` ausgewählten Flag:

```
aws docdb modify-db-cluster \
  --db-cluster-identifier mydocdbcluster \
  --allow-major-version-upgrade \
  --engine-version 5.0.0 \
  --apply-immediately \
  --cluster-parameter-group mydocdbparametergroup \
  --region us-east-1
```

Ersetzen Sie im obigen Beispiel jede *`user input placeholder`* durch die Informationen Ihres Clusters.

Unterschiede zwischen auf Amazon DocumentDB 3.6/4.0 auf 5.0 aktualisierten Clustern und neuen Amazon DocumentDB 5.0-Clustern

- Bei einem direkten Upgrade der Hauptversion werden die ursprünglichen Indizes auf dem aktualisierten Cluster beibehalten. Mit Amazon DocumentDB 5.0 haben wir die Gesamteffizienz der Indexverwaltung und des Garbage-Collection-Prozesses verbessert, insbesondere für Indizes mit niedriger Kardinalität. Als allgemeine bewährte Methode empfehlen wir, Ihre Indizes mit dem Befehl reindex neu zu erstellen, nachdem MVU erfolgreich abgeschlossen wurde. Das Neuerstellen von Indizes ist keine Voraussetzung und erfordert zusätzliche I/O. Weitere Informationen finden Sie unter. Indexverwaltung mit `reIndex`
- Vergleiche von Unterdokumenten für mehrere numerische Datentypen:
  - Wenn der Cluster von Amazon DocumentDB 3.6 migriert wird, erbt er das Verhalten beim Vergleich von Amazon DocumentDB 3.6-Unterdokumenten. Der funktionale Unterschied ist auf numerische Typen (wie Long, Double, Decimal128) in einem Unterdokument beschränkt. `{a: {b: {NumberLong(1)}}`Ist beispielsweise `{a: {b: 1}}` in Amazon DocumentDB 3.6 nicht identisch, während sie in Amazon DocumentDB 4.0 und höher als gleich verglichen werden.
  - Dieses Verhalten beim Vergleich von Unterdokumenten gibt es nur in Amazon DocumentDB 3.6 und in Amazon DocumentDB 5.0-Clustern, die von Version 3.6 mithilfe eines direkten Hauptversions-Upgrades aktualisiert wurden. Dies gilt nicht für neu erstellte Amazon DocumentDB 5.0-Cluster.

> **Note**
>
> Eine Liste der Funktionsunterschiede zwischen Amazon DocumentDB 3.6/4.0 und Amazon DocumentDB 5.0 finden Sie unter. Amazon DocumentDB DocumentDB-Kompatibilität mit MongoDB

Fehlerbehebung bei einem direkten Upgrade der Hauptversion

- Im Falle eines Fehlers versucht das direkte Upgrade der Hauptversion, das Upgrade rückgängig zu machen, um den letzten Betriebszustand des Clusters vor dem Start des Upgrades anzunehmen. Ein erfolgreicher Rollback generiert ein Ereignis: „Der Datenbank-Cluster befindet sich in einem Zustand, der nicht aktualisiert werden kann: Der DocumentDB-Cluster befindet sich in einem Zustand, in dem das Upgrade der Hauptversion nicht erfolgreich abgeschlossen werden kann.“ An diesem Punkt sollten Sie sich an das AWS Support-Team wenden, um das Problem zu beheben und das Versionsupgrade erneut zu versuchen. Sie können Ihren Workload wie bisher weiterverwenden. In allen anderen seltenen Fällen, in denen das Upgrade länger als erwartet dauert, wenden Sie sich bitte an das AWS Support-Team, um Unterstützung zu erhalten.
- Sobald Ihre In-Place-MVU erfolgreich abgeschlossen wurde, kann es bei Ihrem aktualisierten Cluster zu vorübergehenden Leistungseinbußen und zu einer hohen CPU-Auslastung für einen kurzen Zeitraum kommen, während der Aktualisierungsprozess der Index-Metadaten ausgeführt wird. Wenn Sie länger als 2 Stunden weiterhin Leistungseinbußen feststellen, wenden Sie sich bitte an AWS den Support.

Aktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service

> **Important**
>
> Amazon DocumentDB folgt nicht denselben Support-Lebenszyklen wie MongoDB und der end-of-life Zeitplan von MongoDB gilt nicht für Amazon DocumentDB. Derzeit gibt es keine Pläne end-of-life für Amazon DocumentDB 3.6, und Ihre vorhandenen MongoDB 3.6-Treiber, -Anwendungen und -Tools funktionieren weiterhin mit Amazon DocumentDB.

Sie können Ihren Amazon DocumentDB-Cluster mit minimaler Ausfallzeit auf eine höhere Version aktualisieren, indem Sie AWS DMS. AWS DMS ist ein vollständig verwalteter Service, der die Migration von älteren Amazon DocumentDB DocumentDB-Versionen, relationalen Datenbanken und nicht-relationalen Datenbanken zu Ihrem Amazon DocumentDB DocumentDB-Zielcluster erleichtert.

Themen

- Schritt 1: Change Streams aktivieren
- Schritt 2: Ändern Sie die Aufbewahrungsdauer für Change-Streams
- Schritt 3: Migrieren Sie Ihre Indizes
- Schritt 4: Erstellen Sie eine AWS DMS Replikationsinstanz
- Schritt 5: Erstellen Sie einen AWS DMS Quellendpunkt
- Schritt 6: Erstellen Sie einen AWS DMS Zielendpunkt
- Schritt 7: Erstellen Sie eine Migrationsaufgabe und führen Sie sie aus
- Schritt 8: Ändern des Anwendungsendpunkts auf den Amazon DocumentDB DocumentDB-Zielcluster

Schritt 1: Change Streams aktivieren

Um eine Migration mit minimalen Ausfallzeiten durchzuführen, ist Zugriff auf die Change-Streams des Clusters AWS DMS erforderlich. Amazon DocumentDB DocumentDB-Change-Streams bieten eine zeitlich geordnete Abfolge von Aktualisierungsereignissen, die in den Sammlungen und Datenbanken Ihres Clusters auftreten. Das Lesen aus dem Change-Stream ermöglicht es AWS

DMS , Change Data Capture (CDC) durchzuführen und inkrementelle Aktualisierungen auf den Amazon DocumentDB DocumentDB-Zielcluster anzuwenden.

Um Change-Streams für alle Sammlungen in einer bestimmten Datenbank zu aktivieren, authentifizieren Sie sich mit der Mongo-Shell bei Ihrem Amazon DocumentDB-Cluster und führen Sie die folgenden Befehle aus:

```
db.adminCommand({modifyChangeStreams: 1,
    database: "db_name",
    collection: "",
    enable: true});
```

Schritt 2: Ändern Sie die Aufbewahrungsdauer für Change-Streams

Ändern Sie als Nächstes den Aufbewahrungszeitraum für den Change-Stream je nachdem, wie lange Sie Änderungsereignisse im Change-Stream behalten möchten. Wenn Sie beispielsweise davon ausgehen, dass Ihre Amazon AWS DMS DocumentDB-Cluster-Migration 12 Stunden dauern wird, sollten Sie die Aufbewahrung des Change-Streams auf einen Wert von mehr als 12 Stunden festlegen. Die Standardaufbewahrungsdauer für Ihren Amazon DocumentDB-Cluster beträgt drei Stunden. Sie können die Aufbewahrungsdauer des Change-Stream-Protokolls für Ihren Amazon DocumentDB-Cluster zwischen einer Stunde und sieben Tagen ändern, indem Sie die AWS Management Console oder die AWS CLI verwenden. Weitere Informationen finden Sie unter Ändern der Aufbewahrungsdauer für das Change-Stream-Protokoll.

Schritt 3: Migrieren Sie Ihre Indizes

Erstellen Sie dieselben Indizes auf Ihrem Amazon DocumentDB DocumentDB-Zielcluster wie auf Ihrem Amazon DocumentDB DocumentDB-Quellcluster. Es AWS DMS wickelt zwar die Migration von Daten ab, migriert aber keine Indizes. Um die Indizes zu migrieren, verwenden Sie das Amazon DocumentDB Index Tool, um Indizes aus dem Amazon DocumentDB DocumentDB-Quellcluster zu exportieren. Sie können das Tool herunterladen, indem Sie einen Klon des Amazon DocumentDB GitHub DocumentDB-Tools-Repos erstellen und den Anweisungen unter folgen. `README.md` Sie können das Tool von einer EC2 Amazon-Instance oder einer AWS Cloud9 Umgebung aus ausführen, die in derselben Amazon VPC wie Ihr Amazon DocumentDB-Cluster läuft.

Ersetzen Sie im folgenden Beispiel jede Information *`user input placeholder`* durch Ihre eigenen Informationen.

Der folgende Code gibt Indizes aus Ihrem Amazon DocumentDB DocumentDB-Quellcluster aus:

```
python migrationtools/documentdb_index_tool.py --dump-indexes
--uri mongodb://sample-user:user-password@sample-source-cluster.node.us-
east 1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
 21:46:50,432: Successfully connected to instance docdb-40-xx.cluster-xxxxxxxx.us-
east-1.docdb.amazonaws.com:27017
2020-02-11 21:46:50,432: Retrieving indexes from server...2020-02-11 21:46:50,440:
 Completed writing index metadata to local folder: /home/ec2-user/index.js/
```

Sobald Ihre Indizes erfolgreich exportiert wurden, stellen Sie diese Indizes in Ihrem Amazon DocumentDB DocumentDB-Zielcluster wieder her. Verwenden Sie das Amazon DocumentDB Index Tool, um die Indizes wiederherzustellen, die Sie im vorherigen Schritt exportiert haben. Der folgende Befehl stellt die Indizes in Ihrem Amazon DocumentDB DocumentDB-Zielcluster aus dem angegebenen Verzeichnis wieder her.

```
python migrationtools/documentdb_index_tool.py --restore-indexes
--uri mongodb://sample-user:user-password@sample-destination-
cluster.node.us-east 1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
--dir ~/index.js/

2020-02-11 21:51:23,245: Successfully authenticated to database: admin2020-02-11
 21:51:23,245: Successfully connected to instance docdb-50-xx.cluster-xxxxxxxx.us-
east-1.docdb.amazonaws.com:27017
2020-02-11 21:51:23,264: testdb.coll: added index: _id
```

Um zu überprüfen, ob Sie die Indizes korrekt wiederhergestellt haben, stellen Sie mit der Mongo-Shell eine Verbindung zu Ihrem Amazon DocumentDB DocumentDB-Zielcluster her und listen Sie die Indizes für eine bestimmte Sammlung auf. Sehen Sie sich den folgenden Code an:

```
mongo --ssl
--host docdb-xx-xx.cluster-xxxxxxxx.us-east-1.docdb.amazonaws.com:27017
--sslCAFile rds-ca-2019-root.pem --username documentdb --password documentdb

db.coll.getIndexes()
```

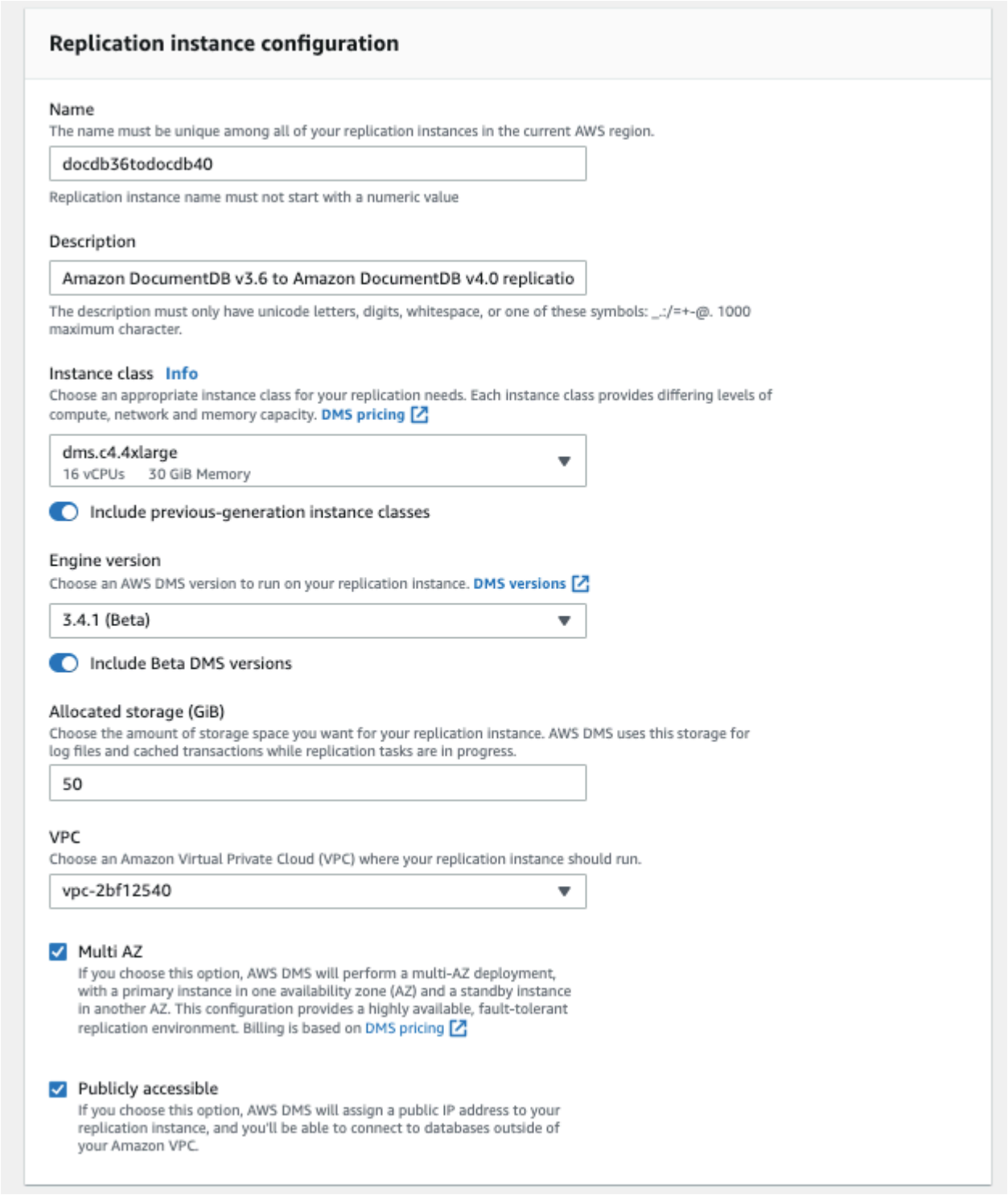
Schritt 4: Erstellen Sie eine AWS DMS Replikationsinstanz

Eine AWS DMS Replikationsinstanz stellt eine Verbindung her und liest Daten aus Ihrem Amazon DocumentDB DocumentDB-Quellcluster und schreibt sie in Ihren Amazon DocumentDB DocumentDB-Zielcluster. Die AWS DMS Replikationsinstanz kann sowohl Massenlade- als auch CDC-Operationen ausführen. Der Großteil dieser Verarbeitung findet im Speicher statt. Bei großen Vorgängen kann jedoch eine gewisse Pufferung auf der Festplatte erforderlich sein. Zwischengespeicherte Transaktionen und Protokolldateien werden ebenfalls auf Festplatte geschrieben. Sobald die Daten migriert sind, streamt die Replikationsinstanz auch alle Änderungsereignisse, um sicherzustellen, dass Quelle und Ziel synchron sind.

So erstellen Sie eine AWS DMS Replikationsinstanz:

1. Öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Replication instances (Replikations-Instances) aus.
3. Wählen Sie Create replication instance (Replikations-Instance erstellen) aus und geben Sie die folgenden Informationen ein:

   - Geben Sie unter Name einen Namen Ihrer Wahl ein. Beispiel, `docdb36todocdb40`.
   - Geben Sie unter Beschreibung eine Beschreibung Ihrer Wahl ein. Für lititem eine Replikationsinstanz von Amazon DocumentDB 3.6 auf Amazon DocumentDB 4.0.
   - Wählen Sie für Instance-Klasse die Größe entsprechend Ihren Anforderungen aus.
   - Wählen Sie für Engine-Version `3.4.1.`
   - Wählen Sie für Amazon VPC die Amazon VPC aus, die Ihre Amazon DocumentDB DocumentDB-Quell- und Zielcluster beherbergt.
   - Verwenden Sie für Allocation Storage (GiB) den Standardwert von 50 GiB. Wenn Sie einen Workload mit hohem Schreibdurchsatz haben, erhöhen Sie diesen Wert entsprechend Ihrer Arbeitslast.
   - Wählen Sie für Multi-AZ Ja aus, wenn Sie Hochverfügbarkeits- und Failover-Unterstützung benötigen.
   - Für Publicly accessible (Öffentlich zugänglich) aktivieren Sie diese Option.



4. Wählen Sie Create replication instance (Replikations-Instance erstellen) aus.

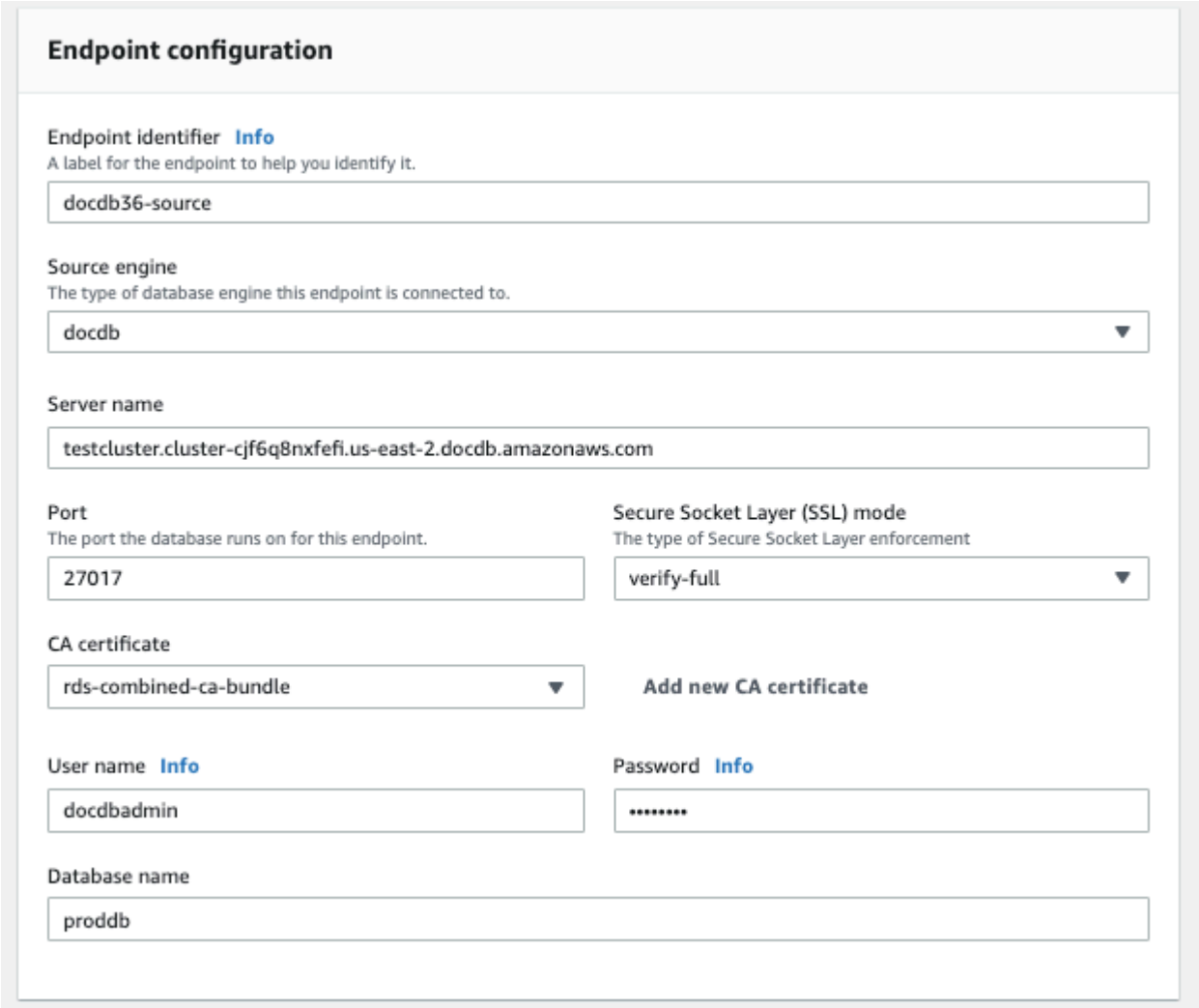
Schritt 5: Erstellen Sie einen AWS DMS Quellendpunkt

Der Quellendpunkt wird für den Amazon DocumentDB DocumentDB-Quellcluster verwendet.

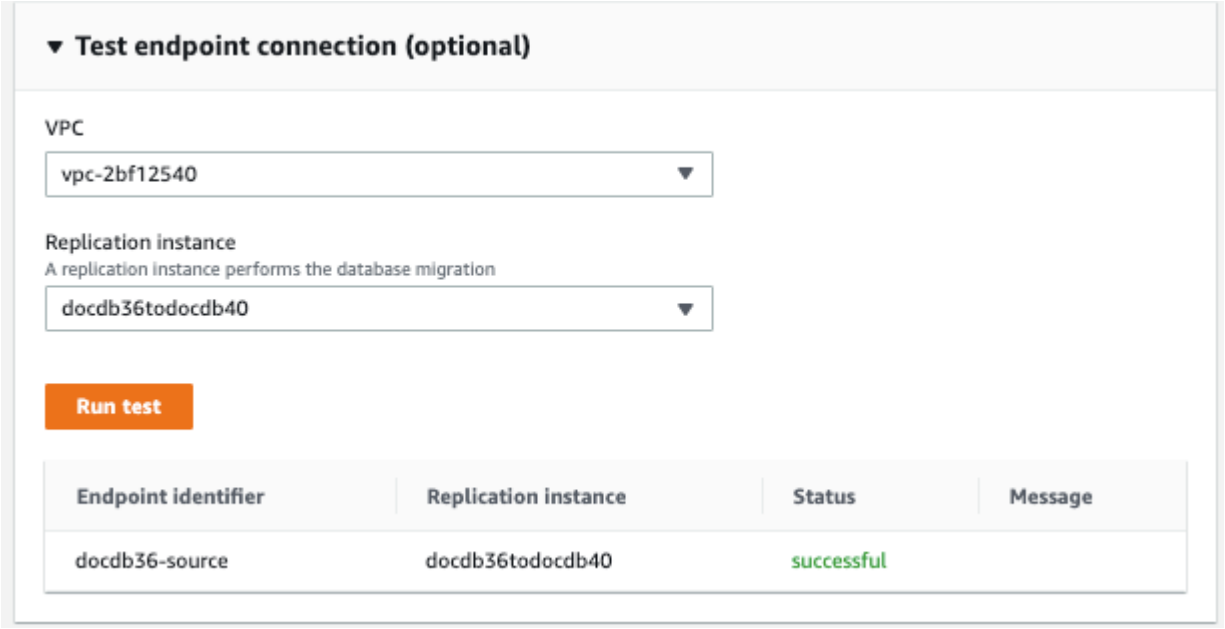
Um einen Quellendpunkt zu erstellen

1. Öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie die folgenden Informationen aus `Create endpoint` und geben Sie sie ein:

   - Für Endpoint type (Endpunkttyp) wählen Sie Source (Quelle) aus.
   - >Geben Sie für Endpoint Identifier beispielsweise `docdb-source` einen Namen ein, den Sie sich leicht merken können.
   - Wählen Sie `docdb` für Source Engine.
   - Geben Sie als Servername den DNS-Namen Ihres Amazon DocumentDB DocumentDB-Quell-Clusters ein.
   - Geben Sie für Port die Portnummer Ihres Amazon DocumentDB DocumentDB-Quell-Clusters ein.
   - Wählen Sie `verify-full` für den SSL-Modus.
   - Wählen Sie für CA-Zertifikat die Option Neues CA-Zertifikat hinzufügen aus. Laden Sie das neue CA-Zertifikat (neues CA-Zertifikat herunter, um ein TLS-Verbindungspaket zu erstellen. Geben Sie als Certificate Identifier ein `rds-combined-ca-bundle`. Wählen Sie unter Import certificate file (Zertifikatsdatei importieren) die Option Choose file (Datei auswählen) und navigieren Sie zu der `.pem`-Datei, die Sie zuvor heruntergeladen haben. Wählen Sie die Datei aus und öffnen Sie sie. Wählen Sie Zertifikat importieren und wählen Sie dann `rds-combined-ca-bundle` aus der Dropdownliste Zertifikat auswählen
   - Geben Sie unter Benutzername den primären Benutzernamen Ihres Amazon DocumentDB DocumentDB-Quell-Clusters ein.
   - Geben Sie unter Passwort das primäre Passwort Ihres Amazon DocumentDB DocumentDB-Quell-Clusters ein.
   - Geben Sie unter Datenbankname den Datenbanknamen ein, für den Sie ein Upgrade durchführen möchten.



4. Testen Sie Ihre Verbindung, um sicherzustellen, dass sie erfolgreich eingerichtet wurde.



5. Klicken Sie auf Endpunkt erstellen.

> **Note**
>
> AWS DMS kann jeweils nur eine Datenbank migrieren.

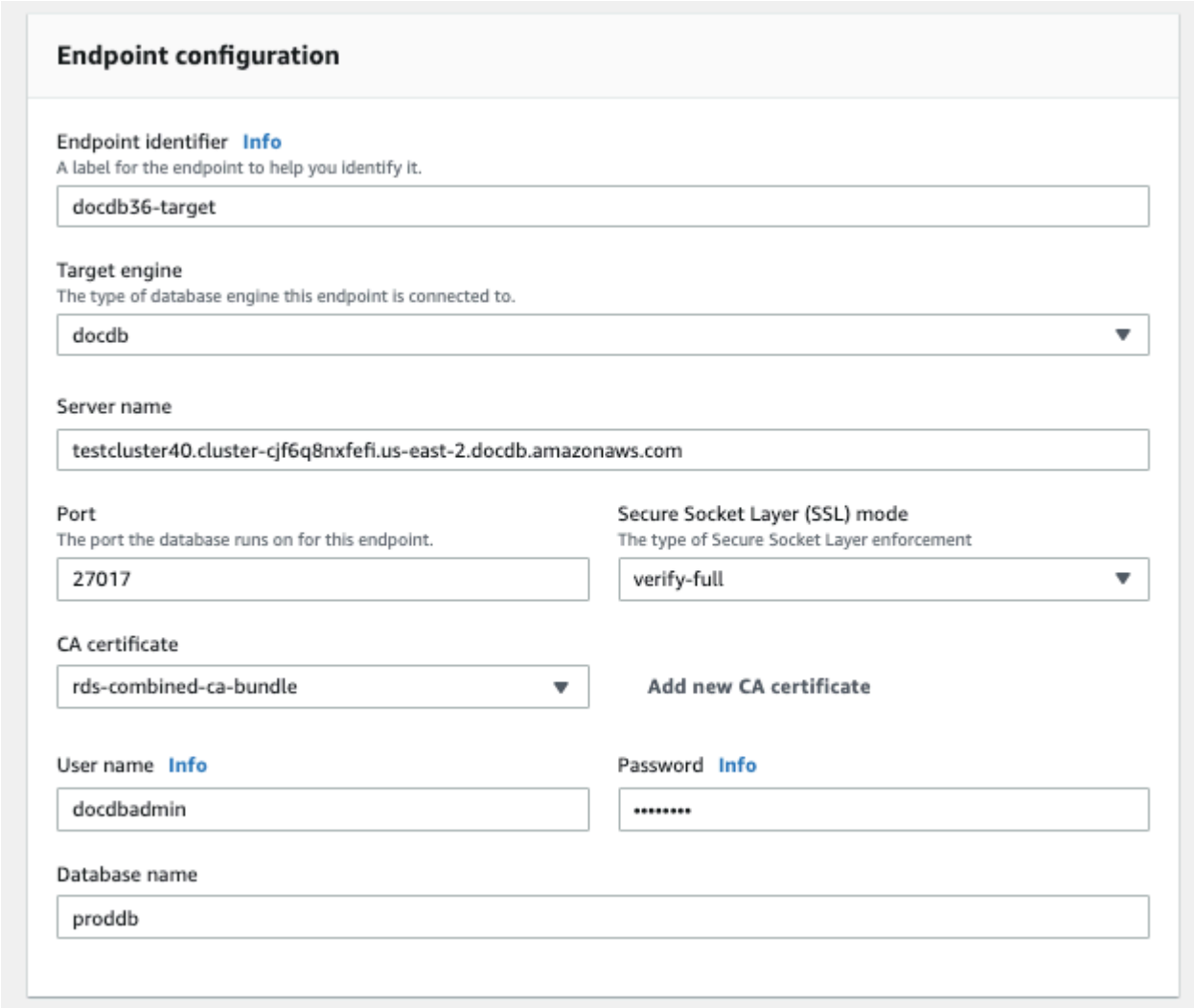
Schritt 6: Erstellen Sie einen AWS DMS Zielendpunkt

Der Zielendpunkt ist für Ihren Amazon DocumentDB DocumentDB-Zielcluster.

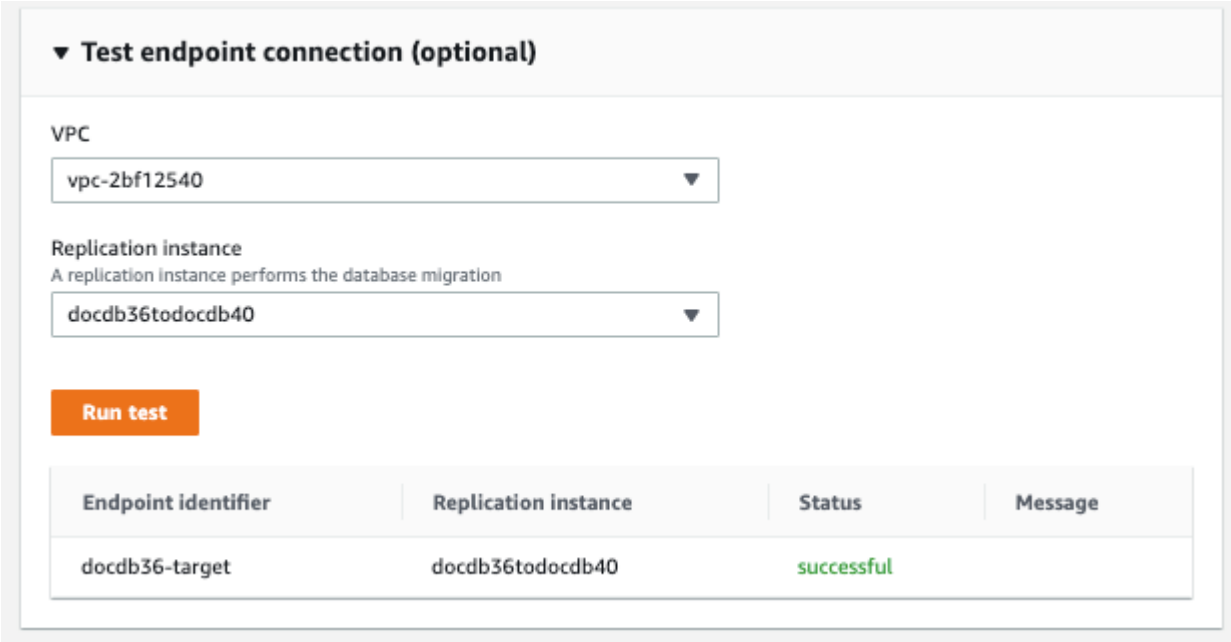
Um einen Zielendpunkt zu erstellen:

1. Öffnen Sie die AWS DMS -Konsole.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Create endpoint (Endpunkt erstellen) und geben Sie die folgenden Informationen ein:
   - Für Endpoint type (Endpunkttyp) wählen Sie Target (Ziel) aus.
   - Für Endpoint Identifier (Endpunkt-ID) geben Sie einen Namen ein, der leicht zu merken ist, zum Beispiel `docdb-target`.
   - Wählen Sie für Source Engine`docdb`.

- Geben Sie als Servername den DNS-Namen Ihres Amazon DocumentDB DocumentDB-Ziel-Clusters ein.
- Geben Sie für Port die Portnummer Ihres Amazon DocumentDB DocumentDB-Ziel-Clusters ein.
- Wählen Sie `verify-full` für den SSL-Modus.
- Wählen Sie für CA-Zertifikat das vorhandene `rds-combined-ca-bundle` Zertifikat aus der Drop-down-Liste Zertifikat auswählen aus.
- Geben Sie unter Benutzername den primären Benutzernamen Ihres Amazon DocumentDB DocumentDB-Ziel-Clusters ein.
- Geben Sie unter Passwort das primäre Passwort Ihres Amazon DocumentDB DocumentDB-Ziel-Clusters ein.
- Geben Sie als Datenbankname denselben Datenbanknamen ein, den Sie zur Einrichtung Ihres Quellendpunkts verwendet haben.



4. Testen Sie Ihre Verbindung, um sicherzustellen, dass sie erfolgreich eingerichtet wurde.



5. Klicken Sie auf Endpunkt erstellen.

Schritt 7: Erstellen Sie eine Migrationsaufgabe und führen Sie sie aus

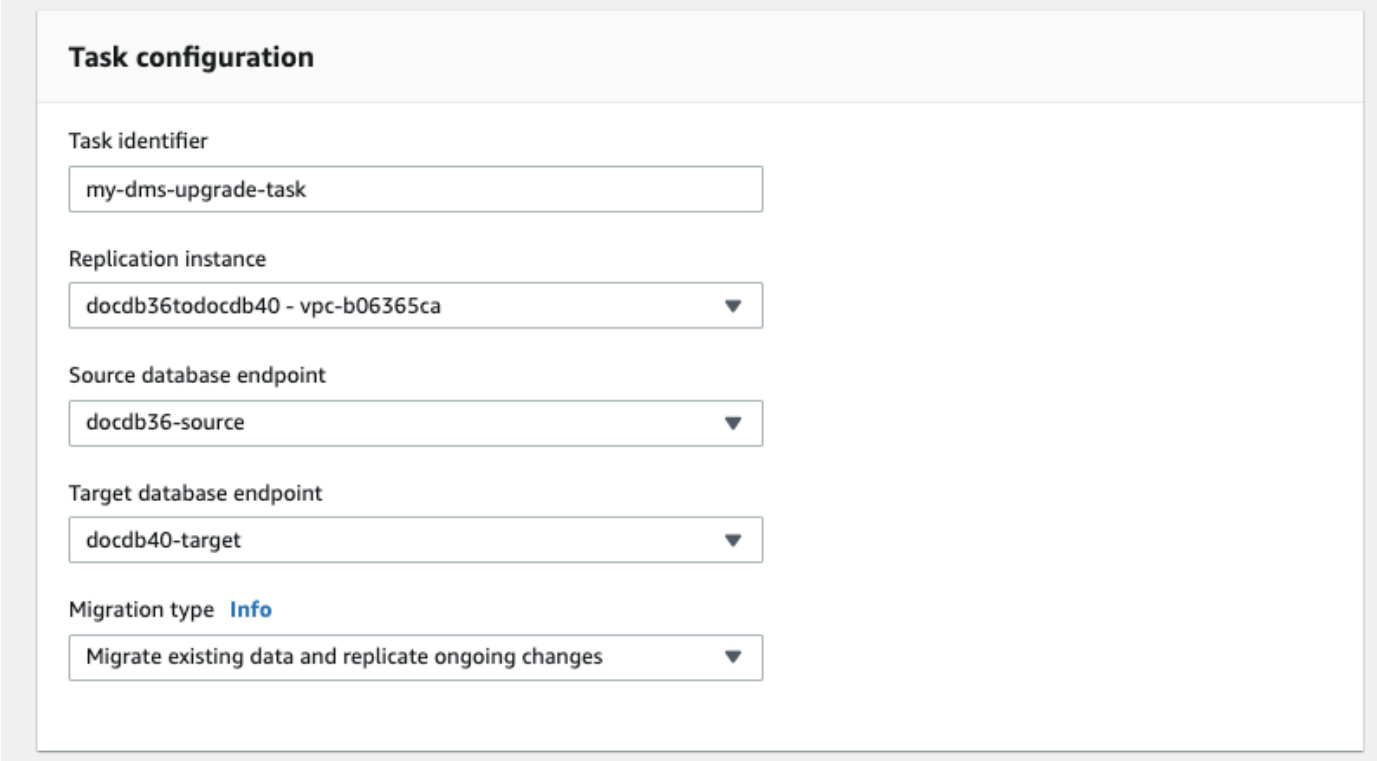
Eine AWS DMS Aufgabe verbindet die Replikationsinstanz mit Ihrer Quell- und Zielinstanz. Wenn Sie eine Migrationsaufgabe erstellen, geben Sie den Quellendpunkt, den Zielendpunkt, die Replikationsinstanz und alle gewünschten Migrationseinstellungen an. Eine AWS DMS Aufgabe kann mit drei verschiedenen Migrationstypen erstellt werden: Migration vorhandener Daten, Migration vorhandener Daten und Replikation laufender Änderungen oder nur Datenänderungen. Da der Zweck dieser Anleitung darin besteht, einen Amazon DocumentDB-Cluster mit minimaler Ausfallzeit zu aktualisieren, nutzen die Schritte die Option, vorhandene Daten zu migrieren und laufende Änderungen zu replizieren. Mit dieser Option werden Änderungen bei der Migration Ihrer vorhandenen Daten AWS DMS erfasst. AWS DMS erfasst weiterhin Änderungen und wendet sie an, auch nachdem die Massendaten geladen wurden. Schließlich sind die Quell- und Zieldatenbanken synchronisiert, unter Berücksichtigung einer minimalen Ausfallzeit während der Migration.

Im Folgenden finden Sie die Schritte zum Erstellen einer Migrationsaufgabe für eine Migration mit minimalen Ausfallzeiten:

1. Öffnen Sie die AWS DMS Konsole.
2. Wählen Sie im Navigationsbereich die Option Datenbankmigrationsaufgaben aus.

3. Wählen Sie Aufgabe zur Datenbankmigration erstellen und geben Sie die folgenden Informationen in den Abschnitt Aufgabenkonfiguration ein:

   - Geben Sie unter Task-ID beispielsweise einen Namen ein, den Sie sich leicht merken können`my-dms-upgrade-task`.
   - Geben Sie für Descriptive Amazon Resource Name (ARN) einen benutzerfreundlichen Namen ein, um den standardmäßigen DMS-ARN zu überschreiben.
   - Für Replication instance (Replikations-Instance) wählen Sie die Replikations-Instance aus, die Sie in Schritt 4: Erstellen Sie eine AWS DMS Replikationsinstanz erstellt haben.
   - Wählen Sie für Quelldatenbank-Endpunkt den Quellendpunkt aus, in dem Sie ihn erstellt haben. Schritt 5: Erstellen Sie einen AWS DMS Quellendpunkt
   - Wählen Sie für Zieldatenbank-Endpunkt den Zielendpunkt aus, in dem Sie ihn erstellt habenSchritt 6: Erstellen Sie einen AWS DMS Zielendpunkt.
   - Wählen Sie als Migrationstyp die Option Migrieren und replizieren aus.

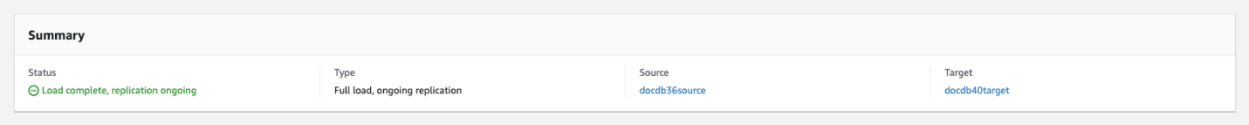


4. Geben Sie im Abschnitt Aufgabeneinstellungen die folgenden Informationen ein:

   - Wählen Sie für den Abschnitt Zieltabellenvorbereitungsmodus die Option Nichts tun aus. Dadurch wird sichergestellt, dass die in Schritt 3 erstellten Indizes nicht gelöscht werden.

- Wählen Sie im Unterabschnitt Task-Protokolle die Option Protokolle aktivieren aus. CloudWatch
- Wählen Sie für die Startkonfiguration der Migrationsaufgabe die Option Automatisch bei Erstellung aus. Dadurch wird die Migrationsaufgabe automatisch gestartet, sobald Sie sie erstellt haben.
- Wählen Sie Datenbankmigrationsaufgabe erstellen aus.

AWS DMS beginnt jetzt mit der Migration von Daten aus Ihrem Amazon DocumentDB DocumentDB-Quellcluster zu Ihrem Amazon DocumentDB DocumentDB-Zielcluster. Der Aufgabenstatus sollte sich von „Wird gestartet“ auf „Wird ausgeführt“ ändern. Sie können den Fortschritt überwachen, indem Sie in der AWS DMS Konsole Aufgaben auswählen. Nach einigen Minuten/Stunden (abhängig vom Umfang Ihrer Migration) sollte sich der Status von „Laden abgeschlossen“ ändern, die Replikation läuft. Das bedeutet, dass AWS DMS Sie eine vollständige Lastmigration Ihres Amazon DocumentDB-Quell-Clusters zu einem Amazon DocumentDB DocumentDB-Zielcluster abgeschlossen haben und jetzt Änderungsereignisse repliziert haben.



Irgendwann werden Ihre Quelle und Ihr Ziel synchronisiert sein. Sie können überprüfen, ob sie synchron sind, indem Sie einen `count()` Vorgang für Ihre Sammlungen ausführen, um zu überprüfen, ob alle Änderungsereignisse migriert wurden.

Schritt 8: Ändern des Anwendungsendpunkts auf den Amazon DocumentDB DocumentDB-Zielcluster

Nachdem der vollständige Ladevorgang abgeschlossen ist und der CDC-Prozess kontinuierlich repliziert wird, können Sie den Datenbankverbindungsendpunkt Ihrer Anwendung von Ihrem Amazon DocumentDB DocumentDB-Quellcluster auf Ihren Amazon DocumentDB DocumentDB-Zielcluster ändern.

Sicherheit in Amazon DocumentDB

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einem Rechenzentrum und einer Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amazon DocumentDB anwenden können. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS -Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon DocumentDB (mit MongoDB-Kompatibilität) gelten, finden Sie unter AWS Services in Scope by Compliance Program.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

Amazon DocumentDB ist im Rahmen des Federal Risk and Authorization Management Program (FedRAMP) autorisiert. Es verfügt über die FedRAMP High-Autorisierung für AWS GovCloud (US-) Regionen und die FedRAMP-Moderat-Autorisierung für AWS die Regionen US-Ost/West. Einzelheiten zu den Compliance-Maßnahmen AWS und den Bemühungen zur Einhaltung der Vorschriften finden Sie unter AWS Services im Umfang nach Compliance-Programmen.

> **Note**
>
> Dieses Kapitel gilt sowohl für instanzbasierte Cluster als auch für Elastic Clusters. Weitere Informationen finden Sie in den folgenden Themen.

Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amazon DocumentDB DocumentDB-Ressourcen überwachen und sichern können. In den folgenden Themen erfahren Sie, wie Sie Amazon DocumentDB konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen.

Themen

- Passwortverwaltung mit Amazon DocumentDB und AWS Secrets Manager
- Datenschutz in Amazon DocumentDB
- Identity and Access Management für Amazon DocumentDB
- Authentifizierung mit IAM-Identität
- Amazon DocumentDB DocumentDB-Benutzer verwalten
- Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle
- Protokollierung und Überwachung in Amazon DocumentDB
- Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate
- Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate — GovCloud
- Konformitätsprüfung in Amazon DocumentDB
- Resilienz in Amazon DocumentDB
- Infrastruktursicherheit in Amazon DocumentDB
- Amazon DocumentDB DocumentDB-API und VPC-Schnittstellen-Endpunkte ()AWS PrivateLink
- Bewährte Sicherheitsmethoden für Amazon DocumentDB
- Amazon DocumentDB DocumentDB-Ereignisse prüfen

Passwortverwaltung mit Amazon DocumentDB und AWS Secrets Manager

Amazon DocumentDB ist in Secrets Manager integriert, um primäre Benutzerkennwörter für Ihre Cluster zu verwalten.

Themen

- Einschränkungen für die Secrets Manager Manager-Integration mit Amazon DocumentDB
- Überblick über die Verwaltung von Primärbenutzerkennwörtern mit AWS Secrets Manager
- Durchsetzung der Amazon DocumentDB DocumentDB-Verwaltung des primären Benutzerkennworts in AWS Secrets Manager
- Verwaltung des primären Benutzerpassworts für einen Cluster mit Secrets Manager

Einschränkungen für die Secrets Manager Manager-Integration mit Amazon DocumentDB

Die Verwaltung von Primärbenutzerkennwörtern mit Secrets Manager wird für die folgenden Funktionen nicht unterstützt:

- Cluster, die Teil einer globalen Amazon DocumentDB DocumentDB-Datenbank sind
- Amazon DocumentDB — Regionsübergreifende Lesereplikate

Überblick über die Verwaltung von Primärbenutzerkennwörtern mit AWS Secrets Manager

Mit AWS Secrets Manager können Sie hartcodierte Anmeldeinformationen in Ihrem Code, einschließlich Datenbankkennwörtern, durch einen API-Aufruf an Secrets Manager ersetzen, um das Geheimnis programmgesteuert abzurufen. Weitere Informationen zu Secrets Manager finden Sie im Benutzerhandbuch für AWS Secrets Manager.

Wenn Sie Datenbankgeheimnisse in Secrets Manager speichern, fallen Gebühren für Ihr AWS Konto an. Informationen zu Preisen finden Sie unter AWS Secrets Manager -Preise.

Sie können angeben, dass Amazon DocumentDB das primäre Benutzerkennwort in Secrets Manager für einen Amazon DocumentDB-Cluster verwaltet, wenn Sie einen der folgenden Vorgänge ausführen:

- Den Cluster erstellen
- Ändern Sie den Cluster

Wenn Sie angeben, dass Amazon DocumentDB das primäre Benutzerkennwort in Secrets Manager verwaltet, generiert Amazon DocumentDB das Passwort und speichert es in Secrets Manager. Sie können direkt mit dem Secret interagieren, um die Anmeldeinformationen für den Hauptbenutzer abzurufen. Sie können auch einen vom Kunden verwalteten Schlüssel angeben, um das Secret zu verschlüsseln, oder den KMS-Schlüssel verwenden, der von Secrets Manager bereitgestellt wird.

Amazon DocumentDB verwaltet die Einstellungen für das Secret und wechselt das Secret standardmäßig alle sieben Tage. Sie können einige Einstellungen ändern, wie zum Beispiel den Rotationsplan. Wenn Sie einen Cluster löschen, der ein Geheimnis in Secrets Manager verwaltet, werden das Geheimnis und die zugehörigen Metadaten ebenfalls gelöscht.

Um eine Verbindung zu einem Cluster mit den Anmeldeinformationen in einem Secret herzustellen, können Sie das Secret von Secrets Manager abrufen. Weitere Informationen finden Sie im AWS Secrets Manager Benutzerhandbuch unter Get Secrets from AWS Secrets Manager and Connect to a SQL database using JDBC with credentials in an AWS Secrets Manager secret.

Durchsetzung der Amazon DocumentDB DocumentDB-Verwaltung des primären Benutzerkennworts in AWS Secrets Manager

Sie können IAM-Bedingungsschlüssel verwenden, um die Amazon DocumentDB DocumentDB-Verwaltung des primären Benutzerkennworts in zu erzwingen. AWS Secrets Manager Die folgende Richtlinie erlaubt es Benutzern nicht, Instances oder Cluster zu erstellen oder wiederherzustellen, es sei denn, das primäre Benutzerkennwort wird von Amazon DocumentDB in Secrets Manager verwaltet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "rds:CreateDBCluster"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "rds:ManageMasterUserPassword": false
                }
            }
        }
    ]
}
```

Verwaltung des primären Benutzerpassworts für einen Cluster mit Secrets Manager

Sie können die Amazon DocumentDB DocumentDB-Verwaltung des primären Benutzerkennworts in Secrets Manager konfigurieren, wenn Sie die folgenden Aktionen ausführen:

- Einen Amazon DocumentDB-Cluster erstellen
- Ändern eines Amazon DocumentDB-Clusters

Sie können die Amazon DocumentDB DocumentDB-Konsole oder die verwenden AWS CLI , um diese Aktionen auszuführen.

Datenschutz in Amazon DocumentDB

Das Modell der AWS gemeinsamen Verantwortung mit gilt für den Datenschutz in Amazon DocumentDB (mit MongoDB-Kompatibilität). Wie in diesem Modell beschrieben, AWS ist es verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen. AWS Cloud 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, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen 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 AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon DocumentDB oder anderen AWS-Services über die Konsole 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

- Clientseitige Verschlüsselung auf Feldebene
- Verschlüsselung von Amazon DocumentDB DocumentDB-Daten im Ruhezustand
- Verschlüsseln von Daten während der Übertragung
- Schlüsselverwaltung

Clientseitige Verschlüsselung auf Feldebene

Mit der clientseitigen Verschlüsselung auf Feldebene (FLE) von Amazon DocumentDB können Sie sensible Daten in Ihren Client-Anwendungen verschlüsseln, bevor sie in einen Amazon DocumentDB-Cluster übertragen werden. Vertrauliche Daten bleiben verschlüsselt, wenn sie in einem Cluster gespeichert und verarbeitet werden, und werden beim Abrufen in der Client-Anwendung entschlüsselt.

Themen

- Erste Schritte
- Abfragen in der clientseitigen FLE
- Einschränkungen

Erste Schritte

Die Erstkonfiguration der clientseitigen FLE in Amazon DocumentDB ist ein vierstufiger Prozess, der die Erstellung eines Verschlüsselungsschlüssels, die Zuordnung einer Rolle zur Anwendung, die Konfiguration der Anwendung und die Definition des CRUD-Vorgangs mit Verschlüsselungsoptionen umfasst.

Themen

- Schritt 1: Erstellen Sie die Verschlüsselungsschlüssel
- Schritt 2: Ordnen Sie der Anwendung eine Rolle zu
- Schritt 3: Konfigurieren Sie die Anwendung
- Schritt 4: Definieren Sie eine CRUD-Operation
- Beispiel: Konfigurationsdatei für die clientseitige Verschlüsselung auf Feldebene

Schritt 1: Erstellen Sie die Verschlüsselungsschlüssel

Erstellen Sie mithilfe AWS Key Management Service eines symmetrischen Schlüssels, der zum Verschlüsseln und Entschlüsseln des vertraulichen Datenfeldes verwendet wird, und erteilen Sie ihm die erforderlichen IAM-Nutzungsberechtigungen. AWS KMS speichert den Kundenschlüssel (CK), der zur Verschlüsselung von Datenschlüsseln () verwendet wird. DKs Wir empfehlen, den Kundenschlüssel in KMS zu speichern, um Ihre Sicherheitslage zu stärken. Der Datenschlüssel ist der Sekundärschlüssel, der in einer Amazon DocumentDB-Sammlung gespeichert wird und benötigt wird, um sensible Felder zu verschlüsseln, bevor das Dokument in Amazon DocumentDB gespeichert wird. Der Kundenschlüssel verschlüsselt den Datenschlüssel, der wiederum Ihre Daten ver- und entschlüsselt. Wenn Sie einen globalen Cluster verwenden, können Sie einen Schlüssel für mehrere Regionen erstellen, der von verschiedenen Servicerollen in verschiedenen Regionen verwendet werden kann.

Weitere Informationen zum AWS Key Management Service, einschließlich der Erstellung eines Schlüssels, finden Sie im AWS Key Management Service Developer Guide.

Schritt 2: Ordnen Sie der Anwendung eine Rolle zu

Erstellen Sie eine IAM-Richtlinie mit den entsprechenden AWS KMS Berechtigungen. Diese Richtlinie ermöglicht es IAM-Identitäten, an die sie angehängt ist, den im Ressourcenfeld angegebenen KMS-Schlüssel zu verschlüsseln und zu entschlüsseln. Ihre Anwendung nimmt diese IAM-Rolle für die Authentifizierung an. AWS KMS

Die Richtlinie sollte etwa wie folgt aussehen:

```
{ "Effect": "Allow",
"Action": ["kms:Decrypt", "kms:Encrypt"],
"Resource": "Customer Key ARN"
}
```

Schritt 3: Konfigurieren Sie die Anwendung

Inzwischen haben Sie einen Kundenschlüssel definiert AWS KMS und eine IAM-Rolle erstellt und ihr die richtigen IAM-Berechtigungen für den Zugriff auf den Kundenschlüssel erteilt. Importieren Sie die erforderlichen Pakete.

```
import boto3
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                                ClientEncryption)
```

```
# create a session object:
my_session = boto3.session.Session()

# get access_key and secret_key programmatically using get_frozen_credentials() method:
 current_credentials = my_session.get_credentials().get_frozen_credentials()
```

1. Geben Sie „aws“ als KMS-Anbietertyp an und geben Sie Ihre Kontoanmeldeinformationen ein, die Sie im vorherigen Schritt abgerufen haben.

   ```
   provider = "aws"
   kms_providers = {
       provider: {
           "accessKeyId": current_credentials.access_key,
           "secretAccessKey": current_credentials.secret_key
       }
   }
   ```

2. Geben Sie den Kundenschlüssel an, der zur Verschlüsselung des Datenschlüssels verwendet wird:

   ```
   customer_key = {
   “region”: “AWS region of the customer_key”,
       “key”: “customer_key ARN”
   }

   key_vault_namespace = "encryption.dataKeys"

   key_alt_name = 'TEST_DATA_KEY'
   ```

3. Konfigurieren Sie das MongoClient Objekt:

```
client = MongoClient(connection_string)

coll = client.test.coll
coll.drop()

client_encryption = ClientEncryption(
    kms_providers, # pass in the kms_providers variable from the previous step
    key_vault_namespace = key_vault_namespace,
    client,
    coll.codec_options
)
```

4. Generieren Sie Ihren Datenschlüssel:

```
data_key_id = client_encryption.create_data_key(provider,
    customer_key,
    key_alt_name = [key_alt_name])
```

5. Rufen Sie Ihren vorhandenen Datenschlüssel ab:

```
data_key = DataKey("aws",
    master_key = customer_key)
key_id = data_key["_id"]
data_key_id = client[key_vault_namespace].find_one({"_id": key_id})
```

Schritt 4: Definieren Sie eine CRUD-Operation

Definieren Sie den CRUD-Vorgang mit Verschlüsselungsoptionen.

1. Definieren Sie die Sammlung in write/read/delete einem einzigen Dokument:

```
coll = client.gameinfo.users
```

2. Explizite Verschlüsselung — Felder verschlüsseln und Folgendes einfügen:

> **Note**
>
> Es muss genau einer der Werte „key_id“ oder „key_alt_name“ angegeben werden.

```
encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_last_name = client_encryption.encrypt(
    "Doe",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)
encrypted_dob = client_encryption.encrypt(
    "1990-01-01",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Random,
    key_alt_name=data_key_id
)

coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name,
    "lastName": encrypted_last_name,
    "dateOfBirth":encrypted_dob,
    "Favorite_games":["Halo","Age of Empires 2","Medal of Honor"]
})
```

Beispiel: Konfigurationsdatei für die clientseitige Verschlüsselung auf Feldebene

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

```
# import python packages:
import boto3
import json
import base64
from pymongo import MongoClient
from pymongo.encryption import (Algorithm,
                                ClientEncryption)

def main():

    # create a session object:
```

```
    my_session = boto3.session.Session()

    # get aws_region from session object:
    aws_region = my_session.region_name

    # get access_key and secret_key programmatically using get_frozen_credentials()
 method:
    current_credentials = my_session.get_credentials().get_frozen_credentials()
    provider = "aws"

    # define the kms_providers which is later used to create the Data Key:
    kms_providers = {
        provider: {
            "accessKeyId": current_credentials.access_key,
            "secretAccessKey": current_credentials.secret_key
        }
    }

    # enter the kms key ARN. Replace the example ARN value.
    kms_arn = "arn:aws:kms:us-east-1:123456789:key/abcd-efgh-ijkl-mnop"
    customer_key = {
        "region": aws_region,
        "key":kms_arn
    }

    # secrets manager is used to strore and retrieve user credentials for connecting to
 an Amazon DocumentDB cluster.
    # retrieve the secret using the secret name. Replace the example secret key.
    secret_name = "/dev/secretKey"
    docdb_credentials = json.loads(my_session.client(service_name = 'secretsmanager',
 region_name = "us-east-1").get_secret_value(SecretId = secret_name)['SecretString'])

    connection_params = '/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false'
    conn_str = 'mongodb://' + docdb_credentials["username"] + ':' +
 docdb_credentials["password"] + '@' + docdb_credentials["host"] + ':' +
 str(docdb_credentials["port"]) + connection_params
    client = MongoClient(conn_str)

    coll = client.test.coll
    coll.drop()

    # store the encryption data keys in a key vault collection (having naming
 convention as db.collection):
```

```
    key_vault_namespace = "encryption.dataKeys"
    key_vault_db_name, key_vault_coll_name = key_vault_namespace.split(".", 1)

    # set up the key vault (key_vault_namespace) for this example:
    key_vault = client[key_vault_db_name][key_vault_coll_name]
    key_vault.drop()
    key_vault.create_index("keyAltNames", unique=True)

    client_encryption = ClientEncryption(
        kms_providers,
        key_vault_namespace,
        client,
        coll.codec_options)

    # create a new data key for the encrypted field:
    data_key_id = client_encryption.create_data_key(provider, master_key=customer_key,
key_alt_names=["some_key_alt_name"], key_material = None)

    # explicitly encrypt a field:
    encrypted_first_name = client_encryption.encrypt(
    "Jane",
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_id=data_key_id
    )
    coll.insert_one(
    {"gamerTag": "jane_doe90",
    "firstName": encrypted_first_name
    })
    doc = coll.find_one()
    print('Encrypted document: %s' % (doc,))

    # explicitly decrypt the field:
    doc["encryptedField"] = client_encryption.decrypt(doc["encryptedField"])
    print('Decrypted document: %s' % (doc,))

    # cleanup resources:
    client_encryption.close()
    client.close()

    if __name__ == "__main__":
        main()
```

Abfragen in der clientseitigen FLE

Amazon DocumentDB unterstützt Punktgleichheitsabfragen mit clientseitiger FLE. Ungleichheits- und Vergleichsabfragen können zu ungenauen Ergebnissen führen. Lese- und Schreibvorgänge können im Vergleich zur Ausführung derselben Operation für den entschlüsselten Wert ein unerwartetes oder falsches Verhalten aufweisen.

Um beispielsweise Filter für Dokumente abzufragen, bei denen der Gamerscore größer als 500 ist:

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
})
```

Der Client verwendet eine explizite Verschlüsselungsmethode, um den Abfragewert zu verschlüsseln:

```
encrypted_gamerscore_filter = client_encryption.encrypt(
    500,
        Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
        key_alt_name=data_key_id
        )

db.users.find( {
    "gamerscore" : { $gt : encrypted_gamerscore_filter }
} )
```

Bei der Suche vergleicht Amazon DocumentDB den verschlüsselten Wert 500 mit den verschlüsselten Feldwerten, die in den einzelnen Dokumenten gespeichert sind, und verwendet dabei die Prüfung „Größer als Ungleichheit“. Die Ungleichheitsprüfung bei der Suche kann zu einem anderen Ergebnis führen, wenn sie mit entschlüsselten Daten und Werten durchgeführt wird, obwohl die Operation erfolgreich Ergebnisse generiert.

Einschränkungen

Die folgenden Einschränkungen gelten für die clientseitige Verschlüsselung auf Feldebene von Amazon DocumentDB:

- Amazon DocumentDB unterstützt nur Punktgleichheitsabfragen. Ungleichheits- und Vergleichsabfragen können zu ungenauen Ergebnissen führen. Lese- und Schreibvorgänge können im Vergleich zur Ausführung derselben Operation für den entschlüsselten Wert ein unerwartetes oder falsches Verhalten aufweisen. Um Filter für Dokumente abzufragen, bei denen der Gamerscore größer als 500 ist.

```
db.users.find( {
    "gamerscore" : { $gt : 500 }
    })
```

Der Client verwendet eine explizite Verschlüsselungsmethode, um den Abfragewert zu verschlüsseln.

```
encrypted_gamerscore_filter = client_encryption.encrypt(
    500,
    Algorithm.AEAD_AES_256_CBC_HMAC_SHA_512_Deterministic,
    key_alt_name=data_key_id
)

db.users.find({
    "gamerscore" : { $gt : encrypted_gamerscore_filter }
})
```

Bei der Suche vergleicht Amazon DocumentDB den verschlüsselten Wert 500 mit den verschlüsselten Feldwerten, die in den einzelnen Dokumenten gespeichert sind, und verwendet dabei die Prüfung „Größer als Ungleichheit“. Die Ungleichheitsprüfung bei der Suche kann zu einem anderen Ergebnis führen, wenn sie mit entschlüsselten Daten und Werten durchgeführt wird, obwohl die Operation erfolgreich Ergebnisse generiert.

- Amazon DocumentDB unterstützt keine explizite clientseitige FLE aus der Mongo Shell. Die Funktion funktioniert jedoch mit jedem unserer unterstützten Treiber.

Verschlüsselung von Amazon DocumentDB DocumentDB-Daten im Ruhezustand

> **Note**
>
> AWS KMS ersetzt den Begriff Customer Master Key (CMK) durch AWS KMS keyeinen KMS-Schlüssel. Das Konzept hat sich nicht geändert. Um bahnbrechende Änderungen zu verhindern, AWS KMS werden einige Varianten dieses Begriffs beibehalten.

Sie verschlüsseln ruhende Daten in Ihrem Amazon DocumentDB-Cluster, indem Sie bei der Erstellung Ihres Clusters die Speicherverschlüsselungsoption angeben. Die Speicher-

Verschlüsselung ist Cluster-weit aktiviert und wird auf alle Instances angewendet, einschließlich der primären Instance und alle Replikate. Sie wird auch auf das Speichervolumen Ihres Clusters, auf Daten, Indizes, Protokolle, automatisierte Sicherungen und Snapshots angewendet.

Amazon DocumentDB verwendet den 256-Bit-Advanced Encryption Standard (AES-256), um Ihre Daten mit den in () gespeicherten Verschlüsselungsschlüsseln zu verschlüsseln. AWS Key Management Service AWS KMS Wenn Sie einen Amazon DocumentDB-Cluster mit aktivierter Verschlüsselung im Ruhezustand verwenden, müssen Sie Ihre Anwendungslogik oder Client-Verbindung nicht ändern. Amazon DocumentDB wickelt die Verschlüsselung und Entschlüsselung Ihrer Daten transparent und mit minimalen Auswirkungen auf die Leistung ab.

Amazon DocumentDB ist in eine Methode integriert, die als Umschlagverschlüsselung bekannt ist, AWS KMS und verwendet diese zum Schutz Ihrer Daten. Wenn ein Amazon DocumentDB-Cluster mit einem verschlüsselt ist AWS KMS, bittet Amazon DocumentDB, Ihren KMS-Schlüssel AWS KMS zu verwenden, um einen Chiffretext-Datenschlüssel zur Verschlüsselung des Speichervolumes zu generieren. Der Chiffretext-Datenschlüssel wird mit dem von Ihnen definierten KMS-Schlüssel verschlüsselt und zusammen mit den verschlüsselten Daten und Speichermetadaten gespeichert. Wenn Amazon DocumentDB auf Ihre verschlüsselten Daten zugreifen muss, fordert AWS KMS es die Entschlüsselung des Chiffretext-Datenschlüssels mithilfe Ihres KMS-Schlüssels an und speichert den Klartext-Datenschlüssel im Speicher, um Daten auf dem Speichervolume effizient zu verschlüsseln und zu entschlüsseln.

Die Speicherverschlüsselungsfunktion in Amazon DocumentDB ist für alle unterstützten Instance-Größen und überall dort verfügbar, AWS-Regionen wo Amazon DocumentDB verfügbar ist.

Verschlüsselung im Ruhezustand für einen Amazon DocumentDB-Cluster aktivieren

Sie können die Verschlüsselung im Ruhezustand auf einem Amazon DocumentDB-Cluster aktivieren oder deaktivieren, wenn der Cluster entweder mit dem AWS Management Console oder dem AWS Command Line Interface ()AWS CLI bereitgestellt wird. Cluster, die Sie mit der Konsole erstellen, haben standardmäßig die Verschlüsselung im Ruhezustand aktiviert. Bei Clustern, die Sie mit dem erstellen, ist AWS CLI die Verschlüsselung im Ruhezustand standardmäßig deaktiviert. Daher müssen Sie die Verschlüsselung im Ruhezustand explizit mit dem `--storage-encrypted`-Parameter aktivieren. In beiden Fällen können Sie nach dem Erstellen des Clusters die Option „Verschlüsselung im Ruhezustand“ nicht ändern.

Amazon DocumentDB verwendet, AWS KMS um Verschlüsselungsschlüssel abzurufen und zu verwalten und die Richtlinien zu definieren, die kontrollieren, wie diese Schlüssel verwendet werden

können. Wenn Sie keine AWS KMS Schlüssel-ID angeben, verwendet Amazon DocumentDB den standardmäßigen KMS-Schlüssel für AWS verwaltete Dienste. Amazon DocumentDB erstellt für jeden AWS-Region in Ihrem AWS-Konto einen eigenen KMS-Schlüssel. Weitere Informationen finden Sie unter AWS Key Management Service -Konzepte.

Informationen zu den ersten Schritten beim Erstellen Ihres eigenen KMS-Schlüssels finden Sie unter Erste Schritte im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie müssen einen KMS-Schlüssel für die symmetrische Verschlüsselung verwenden, um Ihren Cluster zu verschlüsseln, da Amazon DocumentDB nur KMS-Schlüssel mit symmetrischer Verschlüsselung unterstützt. Verwenden Sie keinen asymmetrischen KMS-Schlüssel, um zu versuchen, die Daten in Ihren Amazon DocumentDB-Clustern zu verschlüsseln. Weitere Informationen finden Sie unter Asymmetrische Schlüssel AWS KMS im Entwicklerhandbuch.AWS Key Management Service

Wenn Amazon DocumentDB keinen Zugriff mehr auf den Verschlüsselungsschlüssel für einen Cluster erhält — zum Beispiel, wenn der Zugriff auf einen Schlüssel gesperrt wird — geht der verschlüsselte Cluster in einen Terminalstatus über. In diesem Fall können Sie den Cluster nur aus einer Sicherung wiederherstellen. Für Amazon DocumentDB sind Backups immer für einen Tag aktiviert.

Wenn Sie den Schlüssel für einen verschlüsselten Amazon DocumentDB-Cluster deaktivieren, verlieren Sie außerdem irgendwann den Lese- und Schreibzugriff auf diesen Cluster. Wenn Amazon DocumentDB auf einen Cluster trifft, der mit einem Schlüssel verschlüsselt ist, auf den es keinen Zugriff hat, versetzt es den Cluster in einen Terminalstatus. In diesem Fall ist der Cluster nicht länger verfügbar und der aktuelle Zustand der Datenbank kann nicht mehr wiederhergestellt werden. Um den Cluster wiederherzustellen, müssen Sie den Zugriff auf den Verschlüsselungsschlüssel für Amazon DocumentDB erneut aktivieren und dann den Cluster aus einem Backup wiederherstellen.

> **Important**
>
> Sie können den KMS-Schlüssel für einen verschlüsselten Cluster nicht ändern, nachdem Sie ihn bereits erstellt haben. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel bestimmen, bevor Sie Ihren verschlüsselten Cluster erstellen.

Using the AWS Management Console

Geben Sie beim Erstellen eines Clusters die Option „Verschlüsselung im Ruhezustand“ an. Die Verschlüsselung im Ruhezustand ist standardmäßig aktiviert, wenn Sie einen Cluster mit der AWS Management Console erstellen. Nachdem der Cluster erstellt wurde, kann er nicht mehr geändert werden.

So legen Sie beim Erstellen eines Clusters die Option „Verschlüsselung im Ruhezustand“ fest

1. Erstellen Sie einen Amazon DocumentDB-Cluster, wie im Abschnitt Erste Schritte beschrieben. Wählen Sie jedoch in Schritt 6 nicht Create Cluster (Cluster erstellen) aus.
2. Wählen Sie unterhalb des Abschnitts Authentication (Authentifizierung) die Option Show advanced settings (Erweiterte Einstellungen anzeigen) aus.
3. Scrollen Sie nach unten zum ncryption-at-rest Abschnitt E.
4. Wählen Sie die Option aus, die Sie für die Verschlüsselung im Ruhezustand wünschen. Egal, welche Option Sie wählen, nach dem Erstellen des Clusters können Sie sie nicht mehr ändern.
   - Wenn Sie Daten im Ruhezustand in diesem Cluster verschlüsseln möchten, wählen Sie Enable encryption (Verschlüsselung aktivieren) aus.
   - Wenn Sie Daten im Ruhezustand in diesem Cluster nicht verschlüsseln möchten, wählen Sie Disable encryption (Verschlüsselung deaktivieren) aus.
5. Wählen Sie den gewünschten Primärschlüssel aus. Amazon DocumentDB verwendet die AWS Key Management Service (AWS KMS), um Verschlüsselungsschlüssel abzurufen und zu verwalten und um die Richtlinien zu definieren, die steuern, wie diese Schlüssel verwendet werden können. Wenn Sie keine AWS KMS Schlüssel-ID angeben, verwendet Amazon DocumentDB den standardmäßigen KMS-Schlüssel für AWS verwaltete Dienste. Weitere Informationen finden Sie unter AWS Key Management Service -Konzepte.

   > **Note**
   >
   > Nachdem Sie einen verschlüsselten Cluster erstellt haben, können Sie den KMS-Schlüssel für diesen Cluster nicht ändern. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel bestimmen, bevor Sie Ihren verschlüsselten Cluster erstellen.

6. Füllen Sie die anderen Abschnitte nach Bedarf aus und erstellen Sie Ihren Cluster.

Using the AWS CLI

Um einen Amazon DocumentDB-Cluster mit dem zu verschlüsseln AWS CLI, führen Sie den `create-db-cluster`Befehl aus und geben Sie die `--storage-encrypted` Option an. Amazon DocumentDB-Cluster, die mit dem erstellt wurden, aktivieren standardmäßig AWS CLI keine Speicherverschlüsselung.

Im folgenden Beispiel wird ein Amazon DocumentDB-Cluster mit aktivierter Speicherverschlüsselung erstellt.

Ersetzen Sie in den folgenden Beispielen jedes *`user input placeholder`* durch die Informationen Ihres Clusters.

Example

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
  --db-cluster-identifier mydocdbcluster \
  --port 27017 \
  --engine docdb \
  --master-username SampleUser1 \
  --master-user-password primaryPassword \
  --storage-encrypted
```

Für Windows:

```
aws docdb create-db-cluster ^
  --db-cluster-identifier SampleUser1 ^
  --port 27017 ^
  --engine docdb ^
  --master-username SampleUser1 ^
  --master-user-password primaryPassword ^
  --storage-encrypted
```

Wenn Sie einen verschlüsselten Amazon DocumentDB-Cluster erstellen, können Sie eine AWS KMS Schlüssel-ID angeben, wie im folgenden Beispiel.

Example

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
  --db-cluster-identifier SampleUser1 \
  --port 27017 \
  --engine docdb \
  --master-username primaryUsername \
  --master-user-password yourPrimaryPassword \
  --storage-encrypted \
  --kms-key-id key-arn-or-alias
```

Für Windows:

```
aws docdb create-db-cluster ^
  --db-cluster-identifier SampleUser1 ^
  --port 27017 ^
  --engine docdb ^
  --master-username SampleUser1 ^
  --master-user-password primaryPassword ^
  --storage-encrypted ^
  --kms-key-id key-arn-or-alias
```

> **Note**
>
> Nachdem Sie einen verschlüsselten Cluster erstellt haben, können Sie den KMS-Schlüssel für diesen Cluster nicht ändern. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel bestimmen, bevor Sie Ihren verschlüsselten Cluster erstellen.

Einschränkungen für verschlüsselte Amazon DocumentDB-Cluster

Für verschlüsselte Amazon DocumentDB-Cluster gelten die folgenden Einschränkungen.

- Sie können die Verschlüsselung im Ruhezustand für einen Amazon DocumentDB-Cluster nur zu dem Zeitpunkt aktivieren oder deaktivieren, zu dem er erstellt wurde, nicht nachdem der Cluster erstellt wurde. Sie können jedoch eine verschlüsselte Kopie eines unverschlüsselten Clusters erstellen, indem Sie einen Snapshot des unverschlüsselten Clusters erstellen und dann den unverschlüsselten Snapshot als neuen Cluster wiederherstellen und dabei die Option Verschlüsselung im Ruhezustand angeben.

  Weitere Informationen finden Sie unter den folgenden Themen:

- Einen manuellen Cluster-Snapshot erstellen
- Wiederherstellung aus einem Cluster-Snapshot
- Amazon DocumentDB-Cluster-Snapshots kopieren

- Amazon DocumentDB-Cluster mit aktivierter Speicherverschlüsselung können nicht geändert werden, um die Verschlüsselung zu deaktivieren.
- Alle Instances, automatisierten Backups, Snapshots und Indizes in einem Amazon DocumentDB-Cluster werden mit demselben KMS-Schlüssel verschlüsselt.

Verschlüsseln von Daten während der Übertragung

Sie können Transport Layer Security (TLS) verwenden, um die Verbindung zwischen Ihrer Anwendung und einem Amazon DocumentDB-Cluster zu verschlüsseln. Standardmäßig ist die Verschlüsselung bei der Übertragung für neu erstellte Amazon DocumentDB-Cluster aktiviert. Sie kann bei der Erstellung des Clusters oder zu einem späteren Zeitpunkt optional deaktiviert werden. Wenn die Verschlüsselung während der Übertragung aktiviert ist, sind sichere Verbindungen mit TLS erforderlich, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen zum Herstellen einer Verbindung zu Amazon DocumentDB über TLS finden Sie unter Programmgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB.

TLS-Einstellungen für Amazon DocumentDB-Cluster verwalten

Die Verschlüsselung während der Übertragung für einen Amazon DocumentDB-Cluster wird über den TLS-Parameter in einer Cluster-Parametergruppe verwaltet. Sie können Ihre Amazon DocumentDB-Cluster-TLS-Einstellungen mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) verwalten. In den folgenden Abschnitten finden Sie weitere Informationen zum Überprüfen und Ändern Ihrer aktuellen TLS-Einstellungen.

Using the AWS Management Console

Gehen Sie wie folgt vor, um Verwaltungsaufgaben für die TLS-Verschlüsselung mithilfe der Konsole durchzuführen, z. B. Parametergruppen zu identifizieren, den TLS-Wert zu überprüfen und die erforderlichen Änderungen vorzunehmen.

> **Note**
>
> Sofern Sie beim Erstellen eines Clusters keine andere Angabe machen, wird Ihr Cluster mit der standardmäßigen Cluster-Parametergruppe erstellt. Die Parameter in der

`default`-Cluster-Parametergruppe können nicht geändert werden (z. B. `tls` ist aktiviert/deaktiviert). Wenn Ihr Cluster also eine `default`-Cluster-Parametergruppe verwendet, müssen Sie den Cluster so ändern, dass er eine nicht standardmäßige Cluster-Parametergruppe verwendet. Zuerst müssen Sie möglicherweise eine benutzerdefinierte Cluster-Parametergruppe erstellen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen.

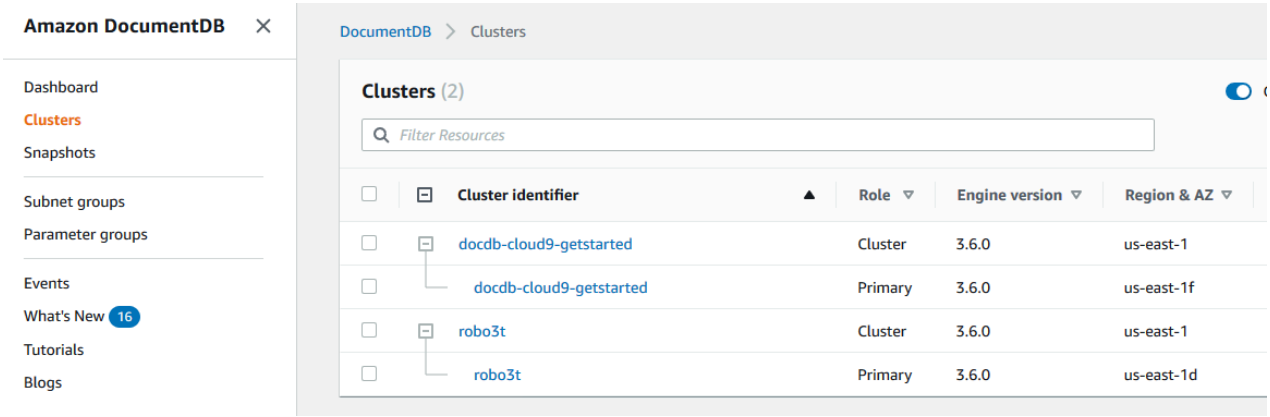
1. Bestimmen Sie, welche Cluster-Parametergruppe Ihr Cluster verwendet.

   a. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

   b. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

   c. Beachten Sie, dass im Cluster-Navigationsfeld in der Spalte Cluster-Identifier sowohl Cluster als auch Instances angezeigt werden. Instances werden unter Clustern aufgeführt. Sehen Sie sich den Screenshot unten als Referenz an.



   d. Wählen Sie den Cluster aus, an dem Sie interessiert sind.

   e. Wählen Sie die Registerkarte Konfiguration und scrollen Sie bis zum Ende der Cluster-Details und suchen Sie die Cluster-Parametergruppe. Der Name der Cluster-Parametergruppe.

Wenn der Name der Cluster-Parametergruppe `default` lautet (z. B. `default.docdb3.6`), müssen Sie eine benutzerdefinierte Cluster-Parametergruppe erstellen und diese zur Parametergruppe des Clusters machen, bevor Sie fortfahren. Weitere Informationen finden Sie hier:

1. Amazon DocumentDB-Cluster-Parametergruppen erstellen— Wenn Sie keine benutzerdefinierte Cluster-Parametergruppe haben, die Sie verwenden können, erstellen Sie eine.
2. Ändern eines Amazon DocumentDB-Clusters— Ändern Sie Ihren Cluster so, dass er die benutzerdefinierte Cluster-Parametergruppe verwendet.

2. Bestimmen Sie den aktuellen Wert des **`tls`**-Cluster-Parameters.

   a. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

   b. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   c. Wählen Sie aus der Liste der Cluster-Parametergruppen den Namen der von Ihnen gewünschten Cluster-Parametergruppe aus.

   d. Suchen Sie den Abschnitt Cluster-Parameter. Suchen Sie in der Liste der Cluster-Parameter die Zeile des `tls`-Cluster-Parameters. An dieser Stelle sind die folgenden vier Spalten wichtig:

      - Cluster-Parametername — Der Name der Cluster-Parameter. Für die Verwaltung von TLS benötigen Sie den `tls`-Cluster-Parameter.
      - Werte — Der aktuelle Wert jedes Cluster-Parameters.
      - Zulässige Werte — Eine Liste von Werten, die auf einen Cluster-Parameter angewendet werden können.
      - Typ anwenden — Entweder statisch oder dynamisch. Änderungen an statischen Cluster-Parametern können nur übernommen werden, wenn die Instances neu gestartet werden. Änderungen an dynamischen Cluster-Parametern können entweder sofort übernommen werden oder wenn die Instances neu gestartet werden.

3. Ändern Sie den Wert des **`tls`**-Cluster-Parameters.

   Wenn der Wert für `tls` nicht der benötigte Wert ist, ändern Sie ihn für diese Cluster-Parametergruppe. Um den Wert des `tls`-Cluster-Parameters zu ändern, fahren Sie nach dem vorherigen Abschnitt mit folgenden Schritten fort.

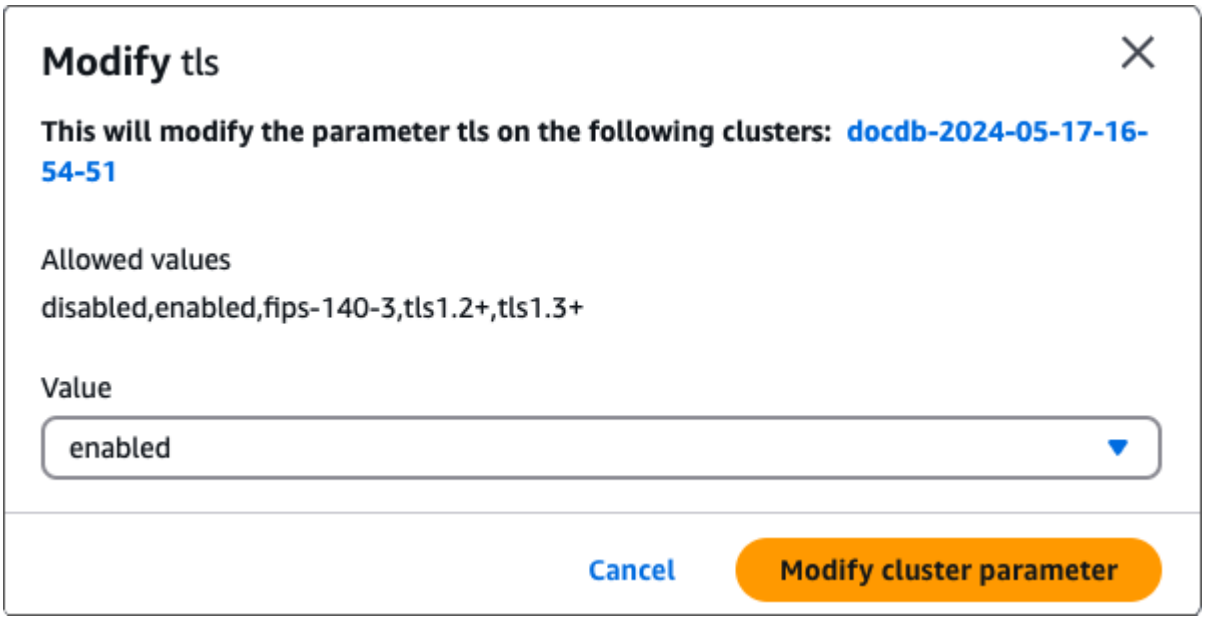
a. Wählen Sie die Schaltfläche links neben dem Namen des Cluster-Parameters (`tls`).

b. Wählen Sie Bearbeiten aus.

c. Um den Wert von zu ändern`tls`, wählen Sie im `tls` Dialogfeld Ändern in der Dropdownliste den gewünschten Wert für den Cluster-Parameter aus.

   Gültige Werte für sind:

   - deaktiviert — Deaktiviert TLS
   - aktiviert — Aktiviert die TLS-Versionen 1.0 bis 1.3.
   - fips-140-3 — Aktiviert TLS mit FIPS. Der Cluster akzeptiert nur sichere Verbindungen gemäß den Anforderungen der Veröffentlichung 140-3 der Federal Information Processing Standards (FIPS). Dies wird erst ab Amazon DocumentDB 5.0-Clustern (Engine-Version 3.0.3727) in diesen Regionen unterstützt: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west
   - tls1.2+ — Aktiviert TLS Version 1.2 und höher. Dies wird erst ab Amazon DocumentDB 4.0 (Engine-Version 2.0.10980) und Amazon DocumentDB (Engine-Version 3.0.11051) unterstützt.
   - tls1.3+ — Aktiviert TLS Version 1.3 und höher. Dies wird erst ab Amazon DocumentDB 4.0 (Engine-Version 2.0.10980) und Amazon DocumentDB (Engine-Version 3.0.11051) unterstützt.



d. Wählen Sie Modify Cluster Parameter (Cluster-Parameter ändern). Die Änderung wird beim Neustart auf jede Cluster-Instance angewendet.

4. Starten Sie die Amazon DocumentDB DocumentDB-Instance neu.

   Starten Sie jede Instance des Clusters neu, sodass die Änderung für alle Instances im Cluster übernommen wird.

   a. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
   b. Wählen Sie im Navigationsbereich Instances aus.
   c. Um eine Instance anzugeben, die neu gestartet werden soll, suchen Sie die Instance in der Liste der Instances und wählen Sie die Schaltfläche links neben dem Namen aus.
   d. Wählen Sie Actions (Aktionen) und dann Reboot (Neustart) aus. Bestätigen Sie, dass Sie neu starten möchten, indem Sie auf Reboot (Neustart) klicken.

Using the AWS CLI

Gehen Sie wie folgt vor, um Verwaltungsaufgaben für die TLS-Verschlüsselung mithilfe von durchzuführen, AWS CLI z. B. Parametergruppen zu identifizieren, den TLS-Wert zu überprüfen und die erforderlichen Änderungen vorzunehmen.

> **Note**
>
> Sofern Sie beim Erstellen eines Clusters keine andere Angabe machen, wird Ihr Cluster mit der standardmäßigen Cluster-Parametergruppe erstellt. Die Parameter in der `default`-Cluster-Parametergruppe können nicht geändert werden (z. B. `tls` ist aktiviert/deaktiviert). Wenn Ihr Cluster also eine `default`-Cluster-Parametergruppe verwendet, müssen Sie den Cluster so ändern, dass er eine nicht standardmäßige Cluster-Parametergruppe verwendet. Möglicherweise müssen Sie zuerst eine benutzerdefinierte Cluster-Parametergruppe erstellen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen.

1. Bestimmen Sie, welche Cluster-Parametergruppe Ihr Cluster verwendet.

   Führen Sie den `describe-db-clusters`Befehl mit den folgenden Optionen aus:

   - `--db-cluster-identifier`

- `--query`

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch die Informationen Ihres Clusters.

```
aws docdb describe-db-clusters \
  --db-cluster-identifier mydocdbcluster \
  --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
[
    [
        "mydocdbcluster",
        "myparametergroup"
    ]
]
```

Wenn der Name der Cluster-Parametergruppe `default` lautet (z. B. `default.docdb3.6`), benötigen Sie eine benutzerdefinierte Cluster-Parametergruppe und müssen diese zur Parametergruppe des Clusters machen, bevor Sie fortfahren. Weitere Informationen finden Sie unter den folgenden Themen:

1. Amazon DocumentDB-Cluster-Parametergruppen erstellen— Wenn Sie keine benutzerdefinierte Cluster-Parametergruppe haben, die Sie verwenden können, erstellen Sie eine.
2. Ändern eines Amazon DocumentDB-Clusters— Ändern Sie Ihren Cluster so, dass er die benutzerdefinierte Cluster-Parametergruppe verwendet.

2. Bestimmen Sie den aktuellen Wert des **`tls`**-Cluster-Parameters.

Um weitere Informationen zu dieser Cluster-Parametergruppe zu erhalten, führen Sie den `describe-db-cluster-parameters`Befehl mit den folgenden Optionen aus:

- `--db-cluster-parameter-group-name`
- `--query`

  Beschränkt die Ausgabe nur auf die Interessenfelder: `ParameterNameParameterValue`,`AllowedValues`, und`ApplyType`.

Ersetzen Sie im folgenden Beispiel jedes *user input placeholder* durch die Informationen Ihres Clusters.

```
aws docdb describe-db-cluster-parameters \
  --db-cluster-parameter-group-name myparametergroup \
  --query 'Parameters[*].[ParameterName,ParameterValue,AllowedValues,ApplyType]'
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
[
    [
        "audit_logs",
        "disabled",
        "enabled,disabled",
        "dynamic"
    ],
    [
        "tls",
        "disabled",
        "disabled,enabled,fips-140-3,tls1.2+,tls1.3+",
        "static"
    ],
    [
        "ttl_monitor",
        "enabled",
        "disabled,enabled",
        "dynamic"
    ]
]
```

3. Ändern Sie den Wert des **`tls`**-Cluster-Parameters.

   Wenn der Wert für `tls` nicht der benötigte Wert ist, ändern Sie ihn für diese Cluster-Parametergruppe. Um den Wert des `tls` Cluster-Parameters zu ändern, führen Sie den `modify-db-cluster-parameter-group`Befehl mit den folgenden Optionen aus:

   - `--db-cluster-parameter-group-name` – Erforderlich. Der Name der zu ändernden Cluster-Parametergruppe. Dabei darf es sich nicht um eine `default.*`-Cluster-Parametergruppe handeln.

- `--parameters` – Erforderlich. Eine Liste der zu ändernden Parameter der Cluster-Parametergruppe.
  - `ParameterName` – Erforderlich. Der Name des zu ändernden Cluster-Parameters.
  - `ParameterValue` – Erforderlich. Der neue Wert für diesen Cluster-Parameter. Muss einer der `AllowedValues` des Cluster-Parameters sein.
    - `enabled`— Der Cluster akzeptiert sichere Verbindungen mit TLS der Versionen 1.0 bis 1.3.
    - `disabled`— Der Cluster akzeptiert keine sicheren Verbindungen mit TLS.
    - `fips-140-3`— Der Cluster akzeptiert nur sichere Verbindungen gemäß den Anforderungen der Veröffentlichung 140-3 der Federal Information Processing Standards (FIPS). Dies wird erst ab Amazon DocumentDB 5.0-Clustern (Engine-Version 3.0.3727) in diesen Regionen unterstützt: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west
    - `tls1.2+`— Der Cluster akzeptiert sichere Verbindungen mit TLS Version 1.2 und höher. Dies wird erst ab Amazon DocumentDB 4.0 (Engine-Version 2.0.10980) und Amazon DocumentDB 5.0 (Engine-Version 3.0.11051) unterstützt.
    - `tls1.3+`— Der Cluster akzeptiert sichere Verbindungen mit TLS Version 1.3 und höher. Dies wird erst ab Amazon DocumentDB 4.0 (Engine-Version 2.0.10980) und Amazon DocumentDB 5.0 (Engine-Version 3.0.11051) unterstützt.
  - `ApplyMethod`— Wann diese Änderung angewendet werden soll. Für statische Cluster-Parameter wie `tle` muss dieser Wert `pending-reboot` lauten.
    - `pending-reboot`— Die Änderung wird erst auf eine Instanz angewendet, nachdem sie neu gestartet wurde. Sie müssen jede Cluster-Instance einzeln neu starten, damit die Änderung für alle Instances des Clusters übernommen wird.

Ersetzen Sie in den folgenden Beispielen jedes Beispiel durch die *`user input placeholder`* Informationen Ihres Clusters.

Der folgende Code deaktiviert `tls` und wendet die Änderung auf jede Instanz an, wenn sie neu gestartet wird.

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name myparametergroup \
  --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

Der folgende Code ermöglicht es `tls` (Version 1.0 bis 1.3), die Änderung auf jede Instanz anzuwenden, wenn diese neu gestartet wird.

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name myparametergroup \
  --parameters "ParameterName=tls,ParameterValue=enabled,ApplyMethod=pending-
reboot"
```

Der folgende Code aktiviert TLS mit `fips-140-3` und wendet die Änderung auf jede Instanz an, wenn sie neu gestartet wird.

```
aws docdb modify-db-cluster-parameter-group \
  --db-cluster-parameter-group-name myparametergroup2 \
  --parameters "ParameterName=tls,ParameterValue=fips-140-3,ApplyMethod=pending-
reboot"
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
{
    "DBClusterParameterGroupName": "myparametergroup"
}
```

4. Starten Sie Ihre Amazon DocumentDB DocumentDB-Instance neu.

   Starten Sie jede Instance des Clusters neu, sodass die Änderung für alle Instances im Cluster übernommen wird. Um eine Amazon DocumentDB DocumentDB-Instance neu zu starten, führen Sie den `reboot-db-instance`Befehl mit der folgenden Option aus:

   - `--db-instance-identifier`

   Der folgende Code startet die Instance `mydocdbinstance` neu.

   Ersetzen Sie in den folgenden Beispielen jedes Beispiel *`user input placeholder`* durch die Informationen Ihres Clusters.

   Example

   Für Linux, macOS oder Unix:

```
aws docdb reboot-db-instance \
  --db-instance-identifier mydocdbinstance
```

Für Windows:

```
aws docdb reboot-db-instance ^
  --db-instance-identifier mydocdbinstance
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
{
    "DBInstance": {
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
        "PendingModifiedValues": {},
        "DBInstanceStatus": "rebooting",
        "DBSubnetGroup": {
            "Subnets": [
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetIdentifier": "subnet-b3806e8f"
                },
                {
```

```
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1d"
                    },
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1f"
                    },
                    "SubnetIdentifier": "subnet-29ab1025"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-91280df6",
            "DBSubnetGroupName": "default"
        },
        "PromotionTier": 2,
        "DBInstanceClass": "db.r5.4xlarge",
        "InstanceCreateTime": "2018-11-05T23:10:49.905Z",
        "PreferredBackupWindow": "00:00-00:30",
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-a50b-44d4-
b6a0-a177d5ff730b",
        "StorageEncrypted": true,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "EngineVersion": "3.6.0",
        "DbiResourceId": "db-SAMPLERESOURCEID",
        "DBInstanceIdentifier": "mydocdbinstance",
        "Engine": "docdb",
        "AvailabilityZone": "us-east-1a",
```

```
        "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-cluster-
instance-00",
        "BackupRetentionPeriod": 1,
        "Endpoint": {
            "Address": "mydocdbinstance.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "Z2R2ITUGPM61AM"
        },
        "DBClusterIdentifier": "mydocdbcluster"
    }
}
```

Es dauert einige Minuten, bis Ihre Instance neu gestartet wird. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Schlüsselverwaltung

Amazon DocumentDB verwendet AWS Key Management Service (AWS KMS), um Verschlüsselungsschlüssel abzurufen und zu verwalten. AWS KMS kombiniert sichere, hochverfügbare Hardware und Software, um ein für die Cloud skaliertes Schlüsselverwaltungssystem bereitzustellen. Mithilfe AWS KMS können Sie Verschlüsselungsschlüssel erstellen und die Richtlinien definieren, die steuern, wie diese Schlüssel verwendet werden können. AWS KMS unterstützt AWS CloudTrail, sodass Sie die Verwendung von Schlüsseln überprüfen können, um sicherzustellen, dass die Schlüssel ordnungsgemäß verwendet werden.

Ihre AWS KMS Schlüssel können in Kombination mit Amazon DocumentDB und unterstützten AWS Diensten wie Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), Amazon Elastic Block Store (Amazon EBS) und Amazon Redshift verwendet werden. Eine Liste der Dienste, die unterstützt werden AWS KMS, finden Sie unter So verwenden Sie AWS Dienste AWS KMS im Entwicklerhandbuch.AWS Key Management Service Weitere Informationen zu AWS KMS finden Sie unter Was ist AWS Key Management Service?

Identity and Access Management für Amazon DocumentDB

AWS Identity and Access Management (IAM) 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 DocumentDB DocumentDB-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 DocumentDB mit IAM
- Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB
- Problembehandlung Amazon DocumentDB DocumentDB-Identität und Zugriff
- Verwaltung der Zugriffsberechtigungen für Ihre Amazon DocumentDB DocumentDB-Ressourcen
- Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für Amazon DocumentDB
- AWS verwaltete Richtlinien für Amazon DocumentDB
- Amazon DocumentDB DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon DocumentDB ausführen.

Servicebenutzer — Wenn Sie den Amazon DocumentDB-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Amazon DocumentDB DocumentDB-Funktionen 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 auf eine Funktion in Amazon DocumentDB nicht zugreifen können, finden Sie weitere Informationen unterProblembehandlung Amazon DocumentDB DocumentDB-Identität und Zugriff.

Service-Administrator — Wenn Sie in Ihrem Unternehmen für Amazon DocumentDB-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon DocumentDB. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon DocumentDB

Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit Amazon DocumentDB verwenden kann, finden Sie unter. So funktioniert Amazon DocumentDB mit IAM

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon DocumentDB zu verwalten. Beispiele für identitätsbasierte Amazon DocumentDB DocumentDB-Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

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 darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter AWS Signature Version 4 für API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. 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 AWS Multi-Faktor-Authentifizierung (MFA) in IAM im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, 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. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens 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 eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung 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 Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) 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 Methoden für die Übernahme einer Rolle 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 (Verbund) 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 zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. 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 verknüpft ist. AWS-Service 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 ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von

Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden.

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. Die Berechtigungen in den Richtlinien legen fest, ob eine 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 heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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 Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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 Inline-Richtlinie wählen, finden Sie unter Auswählen 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.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

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.
- Dienststeuerungsrichtlinien (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 alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter Resource Control Policies (RCPs) 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 DocumentDB mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon DocumentDB zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen mit Amazon DocumentDB verwendet werden können.

IAM-Funktionen, die Sie mit Amazon DocumentDB verwenden können

IAM-Feature	Instanzbasierte Cluster	Elastische Cluster
Identitätsbasierte Richtlinien	Ja	Ja
Ressourcenbasierte Richtlinien	Nein	Nein
Richtlinienaktionen	Ja	Ja
Richtlinienressourcen	Ja	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja	Ja
ACLs	Nein	Nein
ABAC (Tags in Richtlinien)	Teilweise	Ja
Temporäre Anmeldeinformationen	Ja	Ja

IAM-Feature	Instanzbasierte Cluster	Elastische Cluster
Prinzipalberechtigungen	Ja	Ja
Servicerollen	Ja	Ja
Service-verknüpfte Rollen	Nein	Ja

Einen allgemeinen Überblick darüber, wie Amazon DocumentDB und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für Amazon DocumentDB

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Beispiele für identitätsbasierte Amazon DocumentDB DocumentDB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Ressourcenbasierte Richtlinien in Amazon DocumentDB

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienmaßnahmen für Amazon DocumentDB

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Eine Liste der RDS-Aktionen finden Sie unter Von Amazon Relational Database Service definierte Aktionen in der Service Authorization Reference.
> Informationen zu Richtlinienaktionen für elastische Amazon DocumentDB-Cluster finden Sie unter Von Amazon DocumentDB Elastic Clusters definierte Aktionen in der Service Authorization Reference.

Richtlinienaktionen in Amazon DocumentDB verwenden das folgende Präfix vor der Aktion:

```
aws
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "aws:action1",
      "aws:action2"
         ]
```

Beispiele für identitätsbasierte Amazon DocumentDB DocumentDB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Richtlinienressourcen für Amazon DocumentDB

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen

(ARN) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Eine Liste der RDS-Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Von Amazon Relational Database Service definierte Ressourcen in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von Amazon Relational Database Service definierte Aktionen.
> Informationen zu Ressourcentypen für Amazon DocumentDB-Eleastic-Cluster finden Sie unter Ressourcentypen, die von Amazon DocumentDB Elastic Clusters definiert wurden in der Service Authorization Reference.

Beispiele für identitätsbasierte Amazon DocumentDB DocumentDB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Schlüssel für Richtlinienbedingungen für Amazon DocumentDB

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff 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.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Eine Liste der RDS-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon Relational Database Service in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon Relational Database Service definierte Aktionen.
> Bedingungsschlüssel für Amazon DocumentDB Eleastic Clusters finden Sie unter Condition Keys for Amazon DocumentDB Elastic Clusters in der Service Authorization Reference.

Beispiele für identitätsbasierte Amazon DocumentDB DocumentDB-Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

ACLs in Amazon DocumentDB

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon DocumentDB

> **Note**
>
> ABAC wird für instanzbasierte Cluster nur teilweise unterstützt, für elastische Cluster jedoch.

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Temporäre Anmeldeinformationen mit Amazon DocumentDB verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären

Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln von einer Benutzerrolle zu einer IAM-Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für Amazon DocumentDB

Unterstützt Forward Access Sessions (FAS): Ja

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, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service 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.

Servicerollen für Amazon DocumentDB

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von Amazon DocumentDB beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon DocumentDB eine Anleitung dazu bereitstellt.

Servicebezogene Rollen für Amazon DocumentDB

> **Note**
>
> Serviceverknüpfte Rollen werden für instanzbasierte Cluster nicht unterstützt, für elastische Cluster jedoch.

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für Amazon DocumentDB

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon DocumentDB DocumentDB-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) oder ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von Amazon DocumentDB definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Relational Database Service in der Service Authorization Reference.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.
> Informationen zu Richtlinienaktionen für elastische Amazon DocumentDB-Cluster finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon DocumentDB Elastic Clusters in der Service Authorization Reference.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Amazon DocumentDB DocumentDB-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon DocumentDB DocumentDB-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 Ihren Benutzern und Workloads zunächst 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 Richtlinienvalidierung mit 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 Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon DocumentDB DocumentDB-Konsole

Um auf die Amazon DocumentDB DocumentDB-Konsole (mit MongoDB-Kompatibilität) zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon DocumentDB DocumentDB-Ressourcen in Ihrem AWS-Konto aufzulisten und anzuzeigen. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger

ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die Amazon DocumentDB-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die Amazon DocumentDB *ConsoleAccess* - oder *ReadOnly* AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API oder. AWS CLI 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": "*"
        }
    ]
}
```

Problembehandlung Amazon DocumentDB DocumentDB-Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon DocumentDB und IAM auftreten können.

Themen

- Ich bin nicht berechtigt, eine Aktion in Amazon DocumentDB durchzuführen
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff auf meine AWS-Konto Amazon DocumentDB DocumentDB-Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion in Amazon DocumentDB durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `aws:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aws:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `aws:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht 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 DocumentDB 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 DocumentDB auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff auf meine AWS-Konto Amazon DocumentDB DocumentDB-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. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon DocumentDB diese Funktionen unterstützt, finden Sie unterSo funktioniert Amazon DocumentDB 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 unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwaltung der Zugriffsberechtigungen für Ihre Amazon DocumentDB DocumentDB-Ressourcen

Jede AWS Ressource gehört einem AWS-Konto, und die Berechtigungen zum Erstellen oder Zugreifen auf die Ressourcen werden durch Berechtigungsrichtlinien geregelt. Ein Kontoadministrator kann IAM-Identitäten (d. h. Benutzern, Gruppen und Rollen) Berechtigungsrichtlinien zuordnen, und einige Dienste (z. B. AWS Lambda) unterstützen auch das Anhängen von Berechtigungsrichtlinien an Ressourcen.

> **Note**
>
> Ein Kontoadministrator (oder Administratorbenutzer) ist ein Benutzer mit Administratorberechtigungen. Weitere Informationen finden Sie unter Bewährte Methoden für IAM im IAM-Benutzerhandbuch.

Themen

- Ressourcen und Abläufe von Amazon DocumentDB
- Grundlegendes zum Eigentum an Ressourcen
- Verwaltung des Zugriffs auf -Ressourcen
- Spezifizierung von Richtlinienelementen: Aktionen, Auswirkungen, Ressourcen und Prinzipien
- Angeben von Bedingungen in einer Richtlinie

Ressourcen und Abläufe von Amazon DocumentDB

In Amazon DocumentDB ist die primäre Ressource ein Cluster. Amazon DocumentDB unterstützt andere Ressourcen, die mit der primären Ressource verwendet werden können, wie Instances, Parametergruppen und Event-Abonnements. Diese Ressourcen werden als Unterressourcen bezeichnet.

Diesen Ressourcen und Unterressourcen sind eindeutige Amazon-Ressourcennamen (ARNs) zugeordnet, wie in der folgenden Tabelle dargestellt.

Ressourcentyp	ARN-Format
Cluster	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster:` *`db-cluster-name`*
Cluster-Parametergruppe	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-pg:` *`cluster-parameter-group-name`*
Cluster-Snapshot	`arn:aws:rds:` *`region`*`:`*`account-id`* `:cluster-snapshot:` *`cluster-snapshot-name`*
Instance	`arn:aws:rds:` *`region`*`:`*`account-id`* `:db:`*`db-instance-name`*
Sicherheitsgruppe	`arn:aws:rds:` *`region`*`:`*`account-id`* `:secgrp:`*`security-group-name`*
Subnetzgruppe	`arn:aws:rds:` *`region`*`:`*`account-id`* `:subgrp:`*`subnet-group-name`*

Amazon DocumentDB bietet eine Reihe von Vorgängen für die Arbeit mit den Amazon DocumentDB DocumentDB-Ressourcen. Eine Liste der verfügbaren Operationen finden Sie unter Aktionen.

Grundlegendes zum Eigentum an Ressourcen

Ein Ressourcenbesitzer ist derjenige AWS-Konto , der eine Ressource erstellt hat. Das heißt, der Ressourcenbesitzer ist derjenige AWS-Konto der Hauptentität (das Root-Konto, ein IAM-Benutzer oder eine IAM-Rolle), die die Anfrage authentifiziert, mit der die Ressource erstellt wird. Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Wenn Sie Ihre Root-Kontoanmeldedaten verwenden, AWS-Konto um eine Amazon DocumentDB DocumentDB-Ressource, z. B. eine Instance, zu erstellen, sind Sie AWS-Konto der Eigentümer der Amazon DocumentDB DocumentDB-Ressource.
- Wenn Sie in Ihrem einen IAM-Benutzer erstellen AWS-Konto und diesem Benutzer Berechtigungen zum Erstellen von Amazon DocumentDB DocumentDB-Ressourcen erteilen, kann der Benutzer Amazon DocumentDB DocumentDB-Ressourcen erstellen. Ihr, dem der Benutzer angehört AWS-Konto, besitzt jedoch die Amazon DocumentDB DocumentDB-Ressourcen.
- Wenn Sie in Ihrem Unternehmen eine IAM-Rolle AWS-Konto mit den Berechtigungen zum Erstellen von Amazon DocumentDB DocumentDB-Ressourcen erstellen, kann jeder, der diese Rolle übernehmen kann, Amazon DocumentDB DocumentDB-Ressourcen erstellen. Ihr AWS-Konto, zu dem die Rolle gehört, besitzt die Amazon DocumentDB DocumentDB-Ressourcen.

Verwaltung des Zugriffs auf -Ressourcen

Eine Berechtigungsrichtlinie beschreibt, wer Zugriff auf welche Objekte hat. Im folgenden Abschnitt werden die verfügbaren Optionen zum Erstellen von Berechtigungsrichtlinien erläutert.

> **Note**
>
> In diesem Abschnitt wird die Verwendung von IAM im Kontext von Amazon DocumentDB beschrieben. Er enthält keine detaillierten Informationen über den IAM-Service. Eine umfassende IAM-Dokumentation finden Sie unter Was ist IAM? im IAM-Benutzerhandbuch. Informationen zur Syntax und Beschreibungen der IAM-Richtlinien finden Sie unter AWSIAM Policy Reference im IAM-Benutzerhandbuch.

An eine IAM-Identität angefügte Richtlinien werden als identitätsbasierte Richtlinien (IAM-Richtlinien) bezeichnet. An Ressourcen angehängte Richtlinien werden als ressourcenbasierte Richtlinien bezeichnet. Amazon DocumentDB unterstützt nur identitätsbasierte Richtlinien (IAM-Richtlinien).

Themen

- Identitätsbasierte Richtlinien (IAM-Richtlinien)
- Ressourcenbasierte Richtlinien

Identitätsbasierte Richtlinien (IAM-Richtlinien)

Richtlinien können IAM-Identitäten angefügt werden. Sie können z. B. Folgendes tun:

- Hängen Sie eine Berechtigungsrichtlinie an einen Benutzer oder eine Gruppe in Ihrem Konto an — Ein Kontoadministrator kann eine einem bestimmten Benutzer zugeordnete Berechtigungsrichtlinie verwenden, um diesem Benutzer Berechtigungen zum Erstellen einer Amazon DocumentDB DocumentDB-Ressource, z. B. einer Instance, zu erteilen.
- Einer Rolle eine Berechtigungsrichtlinie zuweisen (kontoübergreifende Berechtigungen gewähren) – Sie können einer IAM-Rolle eine identitätsbasierte Berechtigungsrichtlinie zuweisen, um kontoübergreifende Berechtigungen zu erteilen. Ein Administrator kann beispielsweise wie folgt eine Rolle erstellen, um einem anderen AWS-Konto oder einem AWS Dienst kontoübergreifende Berechtigungen zu gewähren:
  1. Der Administrator von Konto A erstellt eine IAM-Rolle und fügt ihr eine Berechtigungsrichtlinie an, die Berechtigungen für Ressourcen in Konto A erteilt.
  2. Der Administrator von Konto A weist der Rolle eine Vertrauensrichtlinie zu, die Konto B als den Prinzipal identifiziert, der die Rolle übernehmen kann.
  3. Der Administrator von Konto B kann dann die Berechtigungen zur Übernahme der Rolle an alle Benutzer in Konto B delegieren. Auf diese Weise können die Benutzer in Konto B Ressourcen in Konto A erstellen oder darauf zugreifen. Der Principal in der Vertrauensrichtlinie kann auch ein AWS Dienstprinzipal sein, wenn Sie einem AWS Dienst die Erlaubnis erteilen möchten, diese Rolle zu übernehmen.

  Weitere Informationen zum Delegieren von Berechtigungen mithilfe von IAM finden Sie unter Zugriffsverwaltung im IAM-Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die es dem Benutzer mit der ID ermöglicht, Instanzen für Sie `123456789012` AWS-Konto zu erstellen. Die neue Instance muss eine Optionsgruppe und eine Parametergruppe verwenden, die mit `default` beginnt, und sie muss die Subnetzgruppe `default` verwenden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
```

```
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:pg:cluster-pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ]
        }
    ]
}
```

Weitere Informationen zur Verwendung identitätsbasierter Richtlinien mit Amazon DocumentDB finden Sie unter. Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für Amazon DocumentDB Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie im Thema Identitäten (Benutzer, Gruppen und Rollen) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Andere Dienste, wie Amazon Simple Storage Service (Amazon S3), unterstützen ebenfalls ressourcenbasierte Berechtigungsrichtlinien. Beispielsweise können Sie einem Amazon S3-Bucket eine Richtlinie zuweisen, um die Zugriffsberechtigungen für diesen Bucket zu verwalten. Amazon DocumentDB unterstützt keine ressourcenbasierten Richtlinien.

Spezifizierung von Richtlinienelementen: Aktionen, Auswirkungen, Ressourcen und Prinzipien

Für jede Amazon DocumentDB DocumentDB-Ressource (siheRessourcen und Abläufe von Amazon DocumentDB) definiert der Service eine Reihe von API-Vorgängen. Weitere Informationen finden Sie unter Aktionen. Um Berechtigungen für diese API-Operationen zu gewähren, definiert Amazon DocumentDB eine Reihe von Aktionen, die Sie in einer Richtlinie angeben können. Für das Durchführen einer API-Operation können Berechtigungen für mehrere Aktionen erforderlich sein.

Grundlegende Richtlinienelemente:

- Ressource – In einer Richtlinie wird der Amazon-Ressourcenname (ARN) zur Identifizierung der Ressource verwendet, für die die Richtlinie gilt.
- Aktion – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenoperationen an, die Sie zulassen oder verweigern möchten. Die `rds:DescribeDBInstances`-Berechtigung erteilt dem Benutzer zum Beispiel Berechtigungen zum Ausführen der `DescribeDBInstances`-Operation.
- Auswirkung – Die von Ihnen festgelegte Auswirkung, wenn der Benutzer die jeweilige Aktion anfordert – entweder „allow“ (Zugriffserlaubnis) oder „deny“ (Zugriffsverweigerung). Wenn Sie den Zugriff auf eine Ressource nicht ausdrücklich gestatten ("Allow"), wird er automatisch verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So können Sie sicherstellen,

dass Benutzer nicht darauf zugreifen können, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.

- Prinzipal – In identitätsbasierten Richtlinien (IAM-Richtlinien) ist der Benutzer, dem die Richtlinie zugewiesen ist, automatisch der Prinzipal. In ressourcenbasierten Richtlinien müssen Sie den Benutzer, das Konto, den Service oder die sonstige Entität angeben, die die Berechtigungen erhalten soll (gilt nur für ressourcenbasierte Richtlinien). Amazon DocumentDB unterstützt keine ressourcenbasierten Richtlinien.

Weitere Informationen zur Syntax und zu Beschreibungen von IAM-Richtlinien finden Sie in der AWS -IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Eine Tabelle mit allen Amazon DocumentDB DocumentDB-API-Aktionen und den Ressourcen, für die sie gelten, finden Sie unterAmazon DocumentDB DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen.

Angeben von Bedingungen in einer Richtlinie

Beim Erteilen von Berechtigungen können Sie mithilfe der IAM-Richtliniensyntax die Bedingungen angeben, unter denen die Richtlinie wirksam werden soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum gilt. Weitere Informationen zum Angeben von Bedingungen in einer Richtliniensyntax finden Sie im Thema Bedingung im IAM Benutzerhandbuch.

Bedingungen werden mithilfe vordefinierter Bedingungsschlüssel formuliert. Amazon DocumentDB hat keine dienstspezifischen Kontextschlüssel, die in einer IAM-Richtlinie verwendet werden können. Eine Liste der globalen Kontextschlüssel für Bedingungen, die für alle Services verfügbar sind, finden Sie unter Verfügbare Schlüssel für Bedingungen im IAM-Benutzerhandbuch.

Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für Amazon DocumentDB

> **Important**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird. Amazon DocumentDB DocumentDB-Konsolen- und API-Aufrufe werden als Aufrufe der Amazon RDS-API protokolliert. AWS CLI Wir empfehlen Ihnen, zunächst die einführenden Themen zu lesen, in denen die grundlegenden Konzepte und Optionen erläutert werden, die Ihnen zur Verwaltung des Zugriffs auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zur Verfügung stehen.

Weitere Informationen finden Sie unter Verwaltung der Zugriffsberechtigungen für Ihre Amazon DocumentDB DocumentDB-Ressourcen.

In diesem Thema finden Sie Beispiele für identitätsbasierte Richtlinien, in denen ein Kontoadministrator den IAM-Identitäten (Benutzer, Gruppen und Rollen) Berechtigungsrichtlinien anfügen kann.

Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateDBInstanceOnly",
            "Effect": "Allow",
            "Action": [
                "rds:CreateDBInstance"
            ],
            "Resource": [
                "arn:aws:rds:*:123456789012:db:test*",
                "arn:aws:rds:*:123456789012:pg:cluster-pg:default*",
                "arn:aws:rds:*:123456789012:subgrp:default"
            ]
        }
    ]
}
```

Die Richtlinie ist ein einzelnes Statement, das die folgenden Berechtigungen für den IAM-Benutzer bestimmt:

- Die Richtlinie ermöglicht es dem IAM-Benutzer, mithilfe der DBInstance Aktion Erstellen eine Instanz zu erstellen (dies gilt auch für den create-db-instance AWS CLI Vorgang und die AWS Management Console).
- Das Element `Resource` gibt an, dass der Benutzer auf oder mit Ressourcen Aktionen ausführen kann. Sie geben Ressourcen über einen Amazon-Ressourcennamen (ARN) an. Dieser ARN enthält den Namen des Dienstes, zu dem die Ressource gehört (`rds`), den AWS-Region (*gibt in diesem Beispiel eine beliebige Region an), die Benutzerkontonummer (123456789012ist in diesem Beispiel die Benutzer-ID) und den Ressourcentyp.

Das `Resource`-Element im Beispiel gibt für den Benutzer die folgenden richtlinienbezogenen Einschränkungen für die Ressourcen an:

- Die Instance-Kennung für die neue Instance muss mit `test` beginnen (zum Beispiel `testCustomerData1`, `test-region2-data`).
- Die Parametergruppe für die neue Instance muss mit `default` beginnen.
- Die Subnetzgruppe für die neue Instance muss mit `default` beginnen.

Das Element `Principal` ist in der Richtlinie nicht angegeben, da in identitätsbasierten Richtlinien die Angabe des Prinzipals als Empfänger der Berechtigung nicht erforderlich ist. Wenn Sie einem Benutzer eine Richtlinie zuweisen, ist der Benutzer automatisch der Prinzipal. Wird die Berechtigungsrichtlinie einer IAM-Rolle angefügt, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen Amazon DocumentDB DocumentDB-API-Vorgängen und den Ressourcen, für die sie gelten, finden Sie unter Amazon DocumentDB DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen.

Für die Verwendung der Amazon DocumentDB DocumentDB-Konsole sind Berechtigungen erforderlich

Damit ein Benutzer mit der Amazon DocumentDB DocumentDB-Konsole arbeiten kann, muss er über Mindestberechtigungen verfügen. Diese Berechtigungen ermöglichen es dem Benutzer, die Amazon DocumentDB DocumentDB-Ressourcen für ihn zu beschreiben AWS-Konto und andere verwandte Informationen bereitzustellen, einschließlich EC2 Amazon-Sicherheits- und Netzwerkinformationen.

Wenn Sie eine IAM-Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Benutzer mit dieser IAM-Richtlinie. Um sicherzustellen, dass diese Benutzer die Amazon DocumentDB DocumentDB-Konsole weiterhin verwenden können, fügen Sie dem Benutzer auch die `AmazonDocDBConsoleFullAccess` verwaltete Richtlinie bei AWS verwaltete Richtlinien für Amazon DocumentDB, wie unter beschrieben.

Sie müssen Benutzern, die nur die Amazon DocumentDB-API AWS CLI oder die Amazon DocumentDB DocumentDB-API aufrufen, keine Mindestberechtigungen für die Konsole gewähren.

Beispiele für vom Kunden verwaltete Richtlinien

In diesem Abschnitt finden Sie Beispielbenutzerrichtlinien, die Berechtigungen für verschiedene Amazon DocumentDB DocumentDB-Aktionen gewähren. Diese Richtlinien funktionieren, wenn

Sie Amazon DocumentDB DocumentDB-API-Aktionen verwenden AWS SDKs, oder die AWS CLI. Bei Verwendung der Konsole müssen Sie zusätzliche konsolenspezifische Berechtigungen erteilen, die im Abschnitt Für die Verwendung der Amazon DocumentDB DocumentDB-Konsole sind Berechtigungen erforderlich erläutert werden.

Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) und Amazon Neptune gemeinsam genutzt wird.

> **Note**
>
> Alle Beispiele verwenden die Region USA Ost (Nord-Virginia) (`us-east-1`) und enthalten ein fiktives Konto. IDs

Beispiele

- Beispiel 1: Erlauben Sie einem Benutzer, eine beliebige Beschreibungsaktion für eine beliebige Amazon DocumentDB DocumentDB-Ressource auszuführen
- Beispiel 2: Verhindern, dass ein Benutzer eine Instance löscht
- Beispiel 3: Verhindern Sie, dass ein Benutzer einen Cluster erstellt, sofern die Speicherverschlüsselung nicht aktiviert ist

Beispiel 1: Erlauben Sie einem Benutzer, eine beliebige Beschreibungsaktion für eine beliebige Amazon DocumentDB DocumentDB-Ressource auszuführen

Die folgende Berechtigungsrichtlinie gewährt Berechtigungen für einen Benutzer, alle Aktionen auszuführen, die mit beginne `Describe`. Diese Aktionen zeigen Informationen über eine Amazon DocumentDB DocumentDB-Ressource, z. B. eine Instance. Das Platzhalterzeichen (*) im `Resource` Element gibt an, dass die Aktionen für alle Amazon DocumentDB DocumentDB-Ressourcen zulässig sind, die dem Konto gehören.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowRDSDescribe",
         "Effect":"Allow",
         "Action":"rds:Describe*",
         "Resource":"*"
```

```
      }
   ]
}
```

Beispiel 2: Verhindern, dass ein Benutzer eine Instance löscht

Die folgenden Berechtigungsrichtlinien erteilen Berechtigungen, um einen Benutzer davon abzuhalten, eine bestimmte Instance zu löschen. Beispielsweise möchten Sie jedem Benutzer, der kein Administrator ist, verbieten, Ihre Produktions-Instances zu löschen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyDelete1",
         "Effect":"Deny",
         "Action":"rds:DeleteDBInstance",
         "Resource":"arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
      }
   ]
}
```

Beispiel 3: Verhindern Sie, dass ein Benutzer einen Cluster erstellt, sofern die Speicherverschlüsselung nicht aktiviert ist

Die folgende Berechtigungsrichtlinie verweigert einem Benutzer die Erlaubnis, einen Amazon DocumentDB-Cluster zu erstellen, sofern die Speicherverschlüsselung nicht aktiviert ist.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PreventUnencryptedDocumentDB",
         "Effect": "Deny",
         "Action": "RDS:CreateDBCluster",
         "Condition": {
         "Bool": {
         "rds:StorageEncrypted": "false"
      },
         "StringEquals": {
         "rds:DatabaseEngine": "docdb"
         }
```

```
      },
      "Resource": "*"
      }
   ]
}
```

AWS verwaltete Richtlinien für Amazon DocumentDB

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie unter AWS Verwaltete Richtlinien im AWS Identity and Access Management-Benutzerhandbuch.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Dienste fügen einer AWS verwalteten Richtlinie gelegentlich zusätzliche Berechtigungen hinzu, um neue Funktionen zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Es ist sehr wahrscheinlich, dass Dienste eine AWS verwaltete Richtlinie aktualisieren, wenn eine neue Funktion eingeführt wird oder wenn neue Operationen verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die `ViewOnlyAccess` AWS verwaltete Richtlinie bietet beispielsweise Lesezugriff auf viele AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und eine Beschreibung der Richtlinien für Jobfunktionen finden Sie unter AWS Verwaltete Richtlinien für Jobfunktionen im AWS Identity and Access Management-Benutzerhandbuch.

Die folgenden AWS verwalteten Richtlinien, die Sie Benutzern in Ihrem Konto zuordnen können, sind spezifisch für Amazon DocumentDB:

- AmazonDocDBFullZugriff— Gewährt vollen Zugriff auf alle Amazon DocumentDB DocumentDB-Ressourcen für das AWS Root-Konto.
- AmazonDocDBReadOnlyAccess— Gewährt schreibgeschützten Zugriff auf alle Amazon DocumentDB DocumentDB-Ressourcen für das Root-Konto. AWS

- AmazonDocDBConsoleFullAccess— Gewährt vollen Zugriff auf die Verwaltung von Amazon DocumentDB- und Amazon DocumentDB Elastic Cluster-Ressourcen mithilfe von. AWS Management Console
- AmazonDocDBElasticReadOnlyAccess— Gewährt schreibgeschützten Zugriff auf alle Amazon DocumentDB Elastic Cluster-Ressourcen für das Root-Konto. AWS
- AmazonDocDBElasticFullAccess— Gewährt vollen Zugriff auf alle Amazon DocumentDB Elastic Cluster-Ressourcen für das AWS Root-Konto.

AmazonDocDBFullZugriff

Diese Richtlinie gewährt Administratorberechtigungen, die einem Principal vollen Zugriff auf alle Amazon DocumentDB DocumentDB-Aktionen gewähren. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Die Amazon DocumentDB DocumentDB-Berechtigungen ermöglichen alle Amazon DocumentDB DocumentDB-Aktionen.
- Einige der EC2 Amazon-Berechtigungen in dieser Richtlinie sind erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch wird sichergestellt, dass Amazon DocumentDB die Ressourcen erfolgreich mit einem Cluster nutzen kann. Die übrigen EC2 Amazon-Berechtigungen in dieser Richtlinie ermöglichen es Amazon DocumentDB, AWS Ressourcen zu erstellen, die erforderlich sind, damit Sie eine Verbindung zu Ihren Clustern herstellen können.
- Die Amazon DocumentDB DocumentDB-Berechtigungen werden bei API-Aufrufen verwendet, um die übergebenen Ressourcen in einer Anfrage zu validieren. Sie sind erforderlich, damit Amazon DocumentDB den übergebenen Schlüssel mit dem Amazon DocumentDB-Cluster verwenden kann.
- Die CloudWatch Protokolle sind erforderlich, damit Amazon DocumentDB sicherstellen kann, dass die Ziele für die Protokollzustellung erreichbar sind und dass sie für die Verwendung von Broker-Protokollen gültig sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
```

```
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:DeleteDBCluster",
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
                "rds:DeleteDBSubnetGroup",
                "rds:DeleteEventSubscription",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
```

```
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Effect": "Allow",
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Action": "iam:CreateServiceLinkedRole",
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWS ServiceName": "rds.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDBReadOnlyAccess

Diese Richtlinie gewährt nur Leseberechtigungen, die es Benutzern ermöglichen, Informationen in Amazon DocumentDB einzusehen. Principals, denen diese Richtlinie beigefügt ist, können weder Aktualisierungen vornehmen oder bestehende Ressourcen löschen, noch können sie neue Amazon DocumentDB DocumentDB-Ressourcen erstellen. Prinzipale mit diesen Berechtigungen können beispielsweise die Liste der Cluster und Konfigurationen, die mit ihrem Konto verknüpft sind, einsehen, aber nicht die Konfiguration oder Einstellungen von Clustern ändern. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Amazon DocumentDB DocumentDB-Berechtigungen ermöglichen es Ihnen, Amazon DocumentDB DocumentDB-Ressourcen aufzulisten, zu beschreiben und Informationen über sie abzurufen.
- EC2 Amazon-Berechtigungen werden verwendet, um die Amazon-VPC, Subnetze und Sicherheitsgruppen zu beschreiben, ENIs die einem Cluster zugeordnet sind.
- Eine Amazon DocumentDB DocumentDB-Berechtigung wird verwendet, um den Schlüssel zu beschreiben, der dem Cluster zugeordnet ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
```

```
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
                "rds:DescribeEvents",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DownloadDBLogFilePortion",
                "rds:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcs"
            ],
            "Effect": "Allow",
            "Resource": "*"
```

```
        },
        {
            "Action": [
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "kms:ListAliases",
                "kms:ListKeyPolicies"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "logs:DescribeLogStreams",
                "logs:GetLogEvents"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/rds/*:log-stream:*",
                "arn:aws:logs:*:*:log-group:/aws/docdb/*:log-stream:*"
            ]
        }
    ]
}
```

AmazonDocDBConsoleFullAccess

Gewährt vollen Zugriff auf die Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen unter Verwendung der folgenden AWS Management Console Optionen:

- Die Amazon DocumentDB-Berechtigungen, um alle Amazon DocumentDB- und Amazon DocumentDB-Cluster-Aktionen zuzulassen.
- Einige der EC2 Amazon-Berechtigungen in dieser Richtlinie sind erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch soll sichergestellt werden, dass Amazon DocumentDB die Ressourcen erfolgreich für die Bereitstellung und Wartung des Clusters nutzen kann. Die restlichen EC2 Amazon-Berechtigungen in dieser Richtlinie ermöglichen es Amazon DocumentDB, AWS Ressourcen zu erstellen, die benötigt werden, damit Sie eine Verbindung zu Ihren Clustern herstellen können, z. VPCEndpoint
- AWS KMS Berechtigungen werden bei API-Aufrufen verwendet, AWS KMS um die übergebenen Ressourcen in einer Anfrage zu validieren. Sie sind erforderlich, damit Amazon DocumentDB

den übergebenen Schlüssel verwenden kann, um die Daten im Ruhezustand mit dem Amazon DocumentDB Elastic Cluster zu verschlüsseln und zu entschlüsseln.

- Die CloudWatch Protokolle sind erforderlich, damit Amazon DocumentDB sicherstellen kann, dass die Ziele für die Protokollzustellung erreichbar sind und dass sie für die Verwendung von Prüfungs- und Profilerstellungsprotokollen gültig sind.
- Secrets Manager Manager-Berechtigungen sind erforderlich, um das angegebene Geheimnis zu validieren und es zu verwenden, um den Admin-Benutzer für Amazon DocumentDB Elastic Clusters einzurichten.
- Amazon RDS-Berechtigungen sind für Amazon DocumentDB-Cluster-Management-Aktionen erforderlich. Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird.
- Mit SNS-Berechtigungen können Principals Abonnements und Themen für Amazon Simple Notification Service (Amazon SNS) abonnieren und Amazon SNS-Nachrichten veröffentlichen.
- Für die Erstellung der serviceverknüpften Rollen, die für die Veröffentlichung von Metriken und Protokollen erforderlich sind, sind IAM-Berechtigungen erforderlich.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DocdbSids",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster",
```

```
                "docdb-elastic:GetPendingMaintenanceAction",
                "docdb-elastic:ListPendingMaintenanceActions",
                "docdb-elastic:ApplyPendingMaintenanceAction",
                "rds:AddRoleToDBCluster",
                "rds:AddSourceIdentifierToSubscription",
                "rds:AddTagsToResource",
                "rds:ApplyPendingMaintenanceAction",
                "rds:CopyDBClusterParameterGroup",
                "rds:CopyDBClusterSnapshot",
                "rds:CopyDBParameterGroup",
                "rds:CreateDBCluster",
                "rds:CreateDBClusterParameterGroup",
                "rds:CreateDBClusterSnapshot",
                "rds:CreateDBInstance",
                "rds:CreateDBParameterGroup",
                "rds:CreateDBSubnetGroup",
                "rds:CreateEventSubscription",
                "rds:CreateGlobalCluster",
                "rds:DeleteDBCluster",
                "rds:DeleteDBClusterParameterGroup",
                "rds:DeleteDBClusterSnapshot",
                "rds:DeleteDBInstance",
                "rds:DeleteDBParameterGroup",
                "rds:DeleteDBSubnetGroup",
                "rds:DeleteEventSubscription",
                "rds:DeleteGlobalCluster",
                "rds:DescribeAccountAttributes",
                "rds:DescribeCertificates",
                "rds:DescribeDBClusterParameterGroups",
                "rds:DescribeDBClusterParameters",
                "rds:DescribeDBClusterSnapshotAttributes",
                "rds:DescribeDBClusterSnapshots",
                "rds:DescribeDBClusters",
                "rds:DescribeDBEngineVersions",
                "rds:DescribeDBInstances",
                "rds:DescribeDBLogFiles",
                "rds:DescribeDBParameterGroups",
                "rds:DescribeDBParameters",
                "rds:DescribeDBSecurityGroups",
                "rds:DescribeDBSubnetGroups",
                "rds:DescribeEngineDefaultClusterParameters",
                "rds:DescribeEngineDefaultParameters",
                "rds:DescribeEventCategories",
                "rds:DescribeEventSubscriptions",
```

```
                "rds:DescribeEvents",
                "rds:DescribeGlobalClusters",
                "rds:DescribeOptionGroups",
                "rds:DescribeOrderableDBInstanceOptions",
                "rds:DescribePendingMaintenanceActions",
                "rds:DescribeValidDBInstanceModifications",
                "rds:DownloadDBLogFilePortion",
                "rds:FailoverDBCluster",
                "rds:ListTagsForResource",
                "rds:ModifyDBCluster",
                "rds:ModifyDBClusterParameterGroup",
                "rds:ModifyDBClusterSnapshotAttribute",
                "rds:ModifyDBInstance",
                "rds:ModifyDBParameterGroup",
                "rds:ModifyDBSubnetGroup",
                "rds:ModifyEventSubscription",
                "rds:ModifyGlobalCluster",
                "rds:PromoteReadReplicaDBCluster",
                "rds:RebootDBInstance",
                "rds:RemoveFromGlobalCluster",
                "rds:RemoveRoleFromDBCluster",
                "rds:RemoveSourceIdentifierFromSubscription",
                "rds:RemoveTagsFromResource",
                "rds:ResetDBClusterParameterGroup",
                "rds:ResetDBParameterGroup",
                "rds:RestoreDBClusterFromSnapshot",
                "rds:RestoreDBClusterToPointInTime"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DependencySids",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "ec2:AllocateAddress",
                "ec2:AssignIpv6Addresses",
                "ec2:AssignPrivateIpAddresses",
                "ec2:AssociateAddress",
```

```
                "ec2:AssociateRouteTable",
                "ec2:AssociateSubnetCidrBlock",
                "ec2:AssociateVpcCidrBlock",
                "ec2:AttachInternetGateway",
                "ec2:AttachNetworkInterface",
                "ec2:CreateCustomerGateway",
                "ec2:CreateDefaultSubnet",
                "ec2:CreateDefaultVpc",
                "ec2:CreateInternetGateway",
                "ec2:CreateNatGateway",
                "ec2:CreateNetworkInterface",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeInstances",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcs",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:ModifyVpcEndpoint",
                "kms:DescribeKey",
                "kms:ListAliases",
                "kms:ListKeyPolicies",
                "kms:ListKeys",
                "kms:ListRetirableGrants",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "sns:ListSubscriptions",
```

```
                "sns:ListTopics",
                "sns:Publish"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "DocdbSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "rds.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DocdbElasticSLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
        }
    ]
}
```

AmazonDocDBElasticReadOnlyAccess

Diese Richtlinie gewährt nur Leseberechtigungen, die es Benutzern ermöglichen, Elastic Cluster-Informationen in Amazon DocumentDB einzusehen. Principals, denen diese Richtlinie beigefügt ist, können weder Aktualisierungen vornehmen oder bestehende Ressourcen löschen, noch können sie neue Amazon DocumentDB DocumentDB-Ressourcen erstellen. Prinzipale mit diesen Berechtigungen können beispielsweise die Liste der Cluster und Konfigurationen, die mit ihrem Konto

verknüpft sind, einsehen, aber nicht die Konfiguration oder Einstellungen von Clustern ändern. Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Mit Amazon DocumentDB-Elastic-Cluster-Berechtigungen können Sie Elastic Cluster-Ressourcen von Amazon DocumentDB auflisten, beschreiben und Informationen über sie abrufen.
- CloudWatch Berechtigungen werden verwendet, um Servicemetriken zu überprüfen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "docdb-elastic:ListClusters",
            "docdb-elastic:GetCluster",
            "docdb-elastic:ListClusterSnapshots",
            "docdb-elastic:GetClusterSnapshot",
            "docdb-elastic:ListTagsForResource"
         ],
         "Resource": "*"
      },
      {
         "Effect": "Allow",
         "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics",
            "cloudwatch:GetMetricStatistics"
         ],
         "Resource": "*"
      }
   ]
}
```

AmazonDocDBElasticFullAccess

Diese Richtlinie gewährt Administratorberechtigungen, die einem Principal vollen Zugriff auf alle Amazon DocumentDB-Aktionen für Amazon DocumentDB Elastic Cluster gewähren.

Diese Richtlinie verwendet AWS Tags (https://docs.aws.amazon.com/tag-editor/latest/userguide/tagging.html) unter bestimmten Bedingungen, um den Zugriff auf Ressourcen einzuschränken. Wenn Sie ein Geheimnis verwenden, muss es mit einem Tag-Schlüssel `DocDBElasticFullAccess`

und einem Tag-Wert gekennzeichnet werden. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden, muss dieser mit einem Tag-Schlüssel `DocDBElasticFullAccess` und einem Tag-Wert versehen werden.

Die Berechtigungen in dieser Richtlinie sind wie folgt gruppiert:

- Elastische Cluster-Berechtigungen von Amazon DocumentDB ermöglichen alle Amazon DocumentDB DocumentDB-Aktionen.
- Einige der EC2 Amazon-Berechtigungen in dieser Richtlinie sind erforderlich, um die übergebenen Ressourcen in einer API-Anfrage zu validieren. Dadurch soll sichergestellt werden, dass Amazon DocumentDB die Ressourcen erfolgreich für die Bereitstellung und Wartung des Clusters nutzen kann. Die übrigen EC2 Amazon-Berechtigungen in dieser Richtlinie ermöglichen es Amazon DocumentDB, AWS Ressourcen zu erstellen, die erforderlich sind, damit Sie sich mit Ihren Clustern wie einem VPC-Endpunkt verbinden können.
- AWS KMS Für Amazon DocumentDB sind Berechtigungen erforderlich, um den übergebenen Schlüssel zum Verschlüsseln und Entschlüsseln der ruhenden Daten innerhalb des Amazon DocumentDB Elastic Clusters verwenden zu können.

  > **Note**
  >
  > Der vom Kunden verwaltete Schlüssel muss über ein Tag mit Schlüssel `DocDBElasticFullAccess` und Tag-Wert verfügen.

- SecretsManager Berechtigungen sind erforderlich, um das angegebene Geheimnis zu validieren und es zu verwenden, um den Admin-Benutzer für Amazon DocumentDB Elastic Clusters einzurichten.

  > **Note**
  >
  > Das verwendete Geheimnis muss ein Tag mit einem Schlüssel `DocDBElasticFullAccess` und einem Tag-Wert haben.

- Für die Erstellung der serviceverknüpften Rollen, die für die Veröffentlichung von Metriken und Protokollen erforderlich sind, sind IAM-Berechtigungen erforderlich.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "DocdbElasticSid",
            "Effect": "Allow",
            "Action": [
                "docdb-elastic:CreateCluster",
                "docdb-elastic:UpdateCluster",
                "docdb-elastic:GetCluster",
                "docdb-elastic:DeleteCluster",
                "docdb-elastic:ListClusters",
                "docdb-elastic:CreateClusterSnapshot",
                "docdb-elastic:GetClusterSnapshot",
                "docdb-elastic:DeleteClusterSnapshot",
                "docdb-elastic:ListClusterSnapshots",
                "docdb-elastic:RestoreClusterFromSnapshot",
                "docdb-elastic:TagResource",
                "docdb-elastic:UntagResource",
                "docdb-elastic:ListTagsForResource",
                "docdb-elastic:CopyClusterSnapshot",
                "docdb-elastic:StartCluster",
                "docdb-elastic:StopCluster",
                "docdb-elastic:GetPendingMaintenanceAction",
                "docdb-elastic:ListPendingMaintenanceActions",
                "docdb-elastic:ApplyPendingMaintenanceAction"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "EC2Sid",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint",
                "ec2:DescribeVpcEndpoints",
                "ec2:DeleteVpcEndpoints",
                "ec2:ModifyVpcEndpoint",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeAvailabilityZones",
                "secretsmanager:ListSecrets"
            ],
```

```
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "KMSSid",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ],
                    "aws:ResourceTag/DocDBElasticFullAccess": "*"
                }
            }
        },
        {
            "Sid": "KMSGrantSid",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/DocDBElasticFullAccess": "*",
                    "kms:ViaService": [
                        "docdb-elastic.*.amazonaws.com"
                    ]
                },
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
```

```
            }
        },
        {
            "Sid": "SecretManagerSid",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:ListSecretVersionIds",
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:GetResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "secretsmanager:ResourceTag/DocDBElasticFullAccess": "*"
                },
                "StringEquals": {
                    "aws:CalledViaFirst": "docdb-elastic.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CloudwatchSid",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "SLRSid",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-
elastic.amazonaws.com/AWSServiceRoleForDocDB-Elastic",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "docdb-elastic.amazonaws.com"
                }
            }
```

```
        }
    ]
}
```

AmazonDocDB- ElasticServiceRolePolicy

Sie können nichts `AmazonDocDBElasticServiceRolePolicy` an Ihre AWS Identity and Access Management Entitäten anhängen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es Amazon DocumentDB ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Serviceverknüpfte Rollen in elastischen Clustern.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
                    ]
                }
            }
        }
    ]
}
```

Amazon DocumentDB DocumentDB-Updates für AWS verwaltete Richtlinien

Änderung	Beschreibung	Datum
AmazonDocDBElasticFullAccess, AmazonDocDBConsoleFullAccess - Änderung	Die Richtlinien wurden aktualisiert, um ausstehende Wartungsaktionen hinzuzufügen.	11.02.2025

Änderung	Beschreibung	Datum
AmazonDocDBElasticFullAccess, AmazonDocDBConsoleFullAccess - Veränderung	Die Richtlinien wurden aktualisiert, um Aktionen zum Starten/Stoppen von Clustern und zum Kopieren von Cluster-Snapshots hinzuzufügen.	21.2.2024
AmazonDocDBElasticReadOnlyAccess, AmazonDocDBElasticFullAccess - Veränderung	Richtlinien wurden aktualisiert, um `cloudwatch:GetMetricData` Aktionen hinzuzufügen.	21.6.2023
AmazonDocDBElasticReadOnlyAccess – Neue Richtlinie	Neue verwaltete Richtlinie für elastische Amazon DocumentDB-Cluster.	08.06.2023
AmazonDocDBElasticFullAccess – Neue Richtlinie	Neue verwaltete Richtlinie für elastische Amazon DocumentDB-Cluster.	05.06.2023
AmazonDocDB- ElasticServiceRolePolicy – Neue Richtlinie	Amazon DocumentDB erstellt eine neue AWS ServiceRoleForDoc DB-Elastic Service-verknüpfte Rolle für elastische Amazon DocumentDB-Cluster.	30.11.2022
AmazonDocDBConsoleFullAccess- Veränderung	Die Richtlinie wurde aktualisiert, um globale und elastische Cluster-Berechtigungen für Amazon DocumentDB hinzuzufügen.	30.11.2022

Änderung	Beschreibung	Datum
AmazonDocDBConsoleFullAccess,AmazonDocDBFullZugriff, AmazonDocDBReadOnlyAccess - Neue Richtlinie	Start des Dienstes.	19.01.2017

Amazon DocumentDB DocumentDB-API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen

Verwenden Sie die folgenden Abschnitte als Referenz, wenn Sie Berechtigungsrichtlinien einrichten Verwendung identitätsbasierter Richtlinien (IAM-Richtlinien) für Amazon DocumentDB und schreiben, die Sie einer IAM-Identität zuordnen können (identitätsbasierte Richtlinien).

Im Folgenden werden alle Amazon DocumentDB DocumentDB-API-Operationen aufgeführt. In der Liste sind die entsprechenden Aktionen enthalten, für die Sie Berechtigungen zur Ausführung der Aktion erteilen können, die AWS Ressource, für die Sie die Berechtigungen erteilen können, und Bedingungsschlüssel, die Sie für eine detaillierte Zugriffskontrolle angeben können. Sie geben die Aktionen im Feld `Action` der Richtlinie, den Ressourcenwert im Feld `Resource` der Richtlinie und die Bedingungen im Feld `Condition` der Richtlinie an. Weitere Informationen über Bedingungen finden Sie unter Angeben von Bedingungen in einer Richtlinie.

Sie können in Ihren Amazon DocumentDB DocumentDB-Richtlinien AWS-weite Bedingungsschlüssel verwenden, um Bedingungen auszudrücken. Eine vollständige Liste der AWS-weiten Schlüssel finden Sie unter Verfügbare Schlüssel im IAM-Benutzerhandbuch.

Sie können IAM-Richtlinien mit dem IAM-Richtliniensimulator testen. Es stellt automatisch eine Liste der Ressourcen und Parameter bereit, die für jede AWS Aktion erforderlich sind, einschließlich Amazon DocumentDB DocumentDB-Aktionen. Der IAM-Richtliniensimulator bestimmt die Berechtigungen, die für jede der von Ihnen angegebenen Aktionen erforderlich sind. Informationen zum IAM-Richtliniensimulator finden Sie unter Testen von IAM-Richtlinien mit dem IAM-Richtliniensimulator im IAM-Benutzerhandbuch.

> **Note**
>
> Um eine Aktion anzugeben, verwenden Sie das Präfix `rds:` gefolgt vom Namen der API-Operation (z. B. `rds:CreateDBInstance`).

Im Folgenden werden Amazon RDS-API-Operationen und die zugehörigen Aktionen, Ressourcen und Bedingungsschlüssel aufgeführt.

Themen

- Amazon DocumentDB DocumentDB-Aktionen, die Berechtigungen auf Ressourcenebene unterstützen
- Amazon DocumentDB DocumentDB-Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen

Amazon DocumentDB DocumentDB-Aktionen, die Berechtigungen auf Ressourcenebene unterstützen

Berechtigungen auf Ressourcenebene bieten die Möglichkeit, die Ressourcen festzulegen, auf denen Benutzer Aktionen ausführen dürfen. Amazon DocumentDB unterstützt teilweise Berechtigungen auf Ressourcenebene. Das bedeutet, dass Sie für bestimmte Amazon DocumentDB DocumentDB-Aktionen steuern können, wann Benutzer diese Aktionen verwenden dürfen, basierend auf Bedingungen, die erfüllt sein müssen, oder auf bestimmten Ressourcen, die Benutzer verwenden dürfen. Beispielsweise können Sie Benutzern die Berechtigung erteilen, nur bestimmte Instances zu ändern.

Im Folgenden werden Amazon DocumentDB DocumentDB-API-Operationen und die zugehörigen Aktionen, Ressourcen und Bedingungsschlüssel aufgeführt.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird. Weitere Amazon DocumentDB DocumentDB-Aktionen und -Berechtigungen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon RDS in der Service Authorization Reference.

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
AddTagsToResource rds:AddTagsToResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ApplyPendingMaintenanceAction rds:ApplyPendingMaintenanceAction	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
Snapshot kopieren DBCluster rds:CopyDBClusterSnapshot	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBCluster rds:CreateDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Cluster-Parametergruppe	rds:cluster-pg-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
	arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
CreateDBClusterParameterGroup rds:CreateDBClusterParameterGroup	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
DBClusterSchnappschuss erstellen rds:CreateDBClusterSnapshot	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
CreateDBInstance rds:CreateDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag
DBSubnetGruppe erstellen rds:CreateDBSubnetGroup	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
LöschenDBInstance rds:DeleteDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
DBSubnetGruppe löschen rds:DeleteDBSubnetGroup	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
Beschreiben DBCluster Parameter Groups rds:DescribeDBClusterParameterGroups	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
Beschreiben Sie die DBCluster Parameter rds:DescribeDBClusterParameters	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
Beschreiben DBClusters rds:DescribeDBClusters	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
Beschreiben DBCluster SnapshotAttributes rds:DescribeDBClusterSnapshotAttributes	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
Beschreiben Sie DBSubnet Gruppen rds:DescribeDBSubnetGroups	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
DescribePendingMaintenanceActions rds:DescribePendingMaintenanceActions	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
Failover DBCluster rds:FailoverDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
ListTagsForResource rds:ListTagsForResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
Modifizieren SieDBCluster rds:ModifyDBCluster	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-name*	rds:cluster-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
Modifizieren SieDBClusterParameterGroup rds:ModifyDBClusterParameterGroup	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
Modifizieren SieDBClusterSnapshotAttribute rds:ModifyDBClusterSnapshotAttribute	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag
Modifizieren SieDBInstance rds:ModifyDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:DatabaseClass rds:db-tag
Starten Sie neu DBInstance rds:RebootDBInstance	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
RemoveTagsFromResource rds:RemoveTagsFromResource	Instance arn:aws:rds: *region*:*account-id* :db:*db-instance-name*	rds:db-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag
ZurücksetzenDBClusterParameterGroup rds:ResetDBClusterParameterGroup	Cluster-Parametergruppe arn:aws:rds: *region*:*account-id* :cluster-pg: *cluster-parameter-group-name*	rds:cluster-pg-tag
WiederherstellungDBClusterFromSnapshot rds:RestoreDBClusterFromSnapshot	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Cluster-Snapshot arn:aws:rds: *region*:*account-id* :cluster-snapshot: *cluster-snapshot-name*	rds:cluster-snapshot-tag

Amazon DocumentDB DocumentDB-API-Operationen und -Aktionen	Ressourcen	Bedingungsschlüssel
WiederherstellungDBClusterToPointInTime rds:RestoreDBClusterToPointInTime	Cluster arn:aws:rds: *region*:*account-id* :cluster: *db-cluster-instance-name*	rds:cluster-tag
	Subnetzgruppe arn:aws:rds: *region*:*account-id* :subgrp:*subnet-group-name*	rds:subgrp-tag

Amazon DocumentDB DocumentDB-Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen

Sie können alle Amazon DocumentDB DocumentDB-Aktionen in einer IAM-Richtlinie verwenden, um Benutzern entweder die Erlaubnis zu erteilen oder zu verweigern, diese Aktion zu verwenden. Allerdings unterstützen nicht alle Amazon DocumentDB DocumentDB-Aktionen Berechtigungen auf Ressourcenebene, mit denen Sie angeben können, für welche Ressourcen eine Aktion ausgeführt werden kann. Die folgenden Amazon DocumentDB DocumentDB-API-Aktionen unterstützen derzeit keine Berechtigungen auf Ressourcenebene. Um diese Aktionen in einer IAM-Richtlinie verwenden zu können, müssen Sie den Benutzern daher die Erlaubnis erteilen, alle Ressourcen für die Aktion zu verwenden, indem Sie einen * Platzhalter für das Element in Ihrer Anweisung verwenden. `Resource`

- `rds:DescribeDBClusterSnapshots`
- `rds:DescribeDBInstances`

Authentifizierung mit IAM-Identität

Amazon DocumentDB-Benutzer und -Anwendungen können IAM-Benutzer und -Rollen verwenden, um sich in einem Amazon DocumentDB-Cluster zu authentifizieren. Die Amazon DocumentDB IAM-Authentifizierung ist eine kennwortlose Authentifizierungsmethode. Außerdem senden Client-Anwendungen die Kennwortgeheimnisse nicht an den Amazon DocumentDB-Cluster, wenn sie

IAM-Rollen/-Benutzer verwenden. Stattdessen werden Client-Verbindungen mithilfe temporärer Sicherheitstoken authentifiziert. AWS STS Benutzer und Anwendungen ohne Administratorrechte können jetzt denselben IAM-Identitäts-RN verwenden, wenn sie ARN Verbindung zu verschiedenen Amazon DocumentDB-Clustern und anderen Diensten herstellen. AWS

Sie können sich auch dafür entscheiden, sowohl die kennwortbasierte als auch die IAM-Authentifizierung zu verwenden, um Benutzer und Anwendungen in einem Amazon DocumentDB-Cluster zu authentifizieren. Die IAM-Authentifizierung ist nur in der instanzbasierten Amazon DocumentDB-Cluster-Version 5.0 verfügbar. Die IAM-Authentifizierung mit IAM-Identität ARNs wird für den Amazon DocumentDB DocumentDB-Hauptbenutzer nicht unterstützt.

> **Note**
>
> Der Hauptbenutzer kann nur mithilfe einer vorhandenen passwortbasierten Authentifizierung authentifiziert werden.

Themen

- Erste Schritte mit der Authentifizierung mithilfe von IAM-Benutzern und -Rollen
- Konfiguration von AWS Compute-Typen für die Authentifizierung bei Amazon DocumentDB mithilfe von IAM AWS
- Überwachung von IAM-Authentifizierungsanfragen
- Verwenden der IAM-Authentifizierung
- Treiber, die IAM unterstützen
- Häufig gestellte Fragen zur IAM-Identitätsauthentifizierung

Erste Schritte mit der Authentifizierung mithilfe von IAM-Benutzern und -Rollen

Amazon DocumentDB DocumentDB-Benutzer und -Rollen mit IAM-Identitäten werden in einer Datenbank erstellt und verwaltet. `$external`

Einen Benutzer erstellen

Connect als Hauptbenutzer her und erstellen Sie dann einen IAM-Benutzer und eine IAM-Rolle:

```
use $external;
```

```
db.createUser(
    {
        user: "arn:aws:iam::123456789123:user/iamuser",
        mechanisms: ["MONGODB-AWS"],
        roles: [ { role: "readWrite", db: "readWriteDB" } ]
    }
);
```

Alternativ können Sie einen Amazon DocumentDB DocumentDB-Benutzer mit einer IAM-Rolle hinzufügen:

```
use $external;
db.createUser(
    {
        user: "arn:aws:iam::123456789123:role/iamrole",
        mechanisms: ["MONGODB-AWS"],
        roles: [ { role: "readWrite", db: "readWriteDB" } ]
    }
);
```

Ändern eines IAM-Benutzers oder einer IAM-Rolle

Ändern Sie einen vorhandenen IAM-Benutzer:

```
use $external;
db.updateUser(
  "arn:aws:iam::123456789123:user/iamuser",
  {
    roles: [ { role: "read", db: "readDB" } ]
  }
);
```

Ändern Sie eine bestehende IAM-Rolle:

```
use $external;
db.updateUser(
  "arn:aws:iam::123456789123:role/iamrole",
  {
    roles: [ { role: "read", db: "readDB" } ]
  }
);
```

So gewähren oder entziehen Sie einem IAM-Benutzer Rollen:

```
use $external;
db.grantRolesToUser(
  "arn:aws:iam::123456789123:user/iamuser",
  [ { db: "admin", role: "readWriteAnyDatabase" } ]
);
```

```
use $external;
db.revokeRolesFromUser(
  "arn:aws:iam::123456789123:user/iamuser",
  [ { db: "admin", role: "readWriteAnyDatabase" } ]
);
```

So gewähren oder entziehen Sie Rollen für eine IAM-Rolle:

```
use $external;
db.grantRolesToUser(
  "arn:aws:iam::123456789123:user/iamrole",
  [ { db: "admin", role: "readWriteAnyDatabase" } ]
);
```

```
use $external;
db.revokeRolesFromUser(
  "arn:aws:iam::123456789123:user/iamrole",
  [ { db: "admin", role: "readWriteAnyDatabase" } ]
);
```

Löschen eines IAM-Benutzers oder einer IAM-Rolle

Um einen vorhandenen IAM-Benutzer zu löschen:

```
use $external;
db.dropUser("arn:aws:iam::123456789123:user/iamuser");
```

Um eine bestehende IAM-Rolle zu löschen:

```
use $external;
db.dropUser("arn:aws:iam::123456789123:role/iamrole");
```

Konfigurieren Sie einen Verbindungs-URI für die Authentifizierung mit IAM AWS

Verwenden Sie die folgenden URI-Parameter, um sich mit AWS IAM zu authentifizieren: `authSource` as und as. `$external` `authMechanism` `MONGODB-AWS` Wenn Sie einen IAM-Benutzer verwenden, werden die Felder für den Benutzernamen und das Passwort durch einen Zugriffsschlüssel bzw. einen geheimen Schlüssel ersetzt. Wenn Sie eine IAM-Rolle übernehmen, die der Umgebung zugeordnet ist, in der Sie sich befinden (z. B. AWS Lambda Funktion, EC2 Amazon-Instance). Sie müssen bei der Authentifizierung mit diesem Mechanismus keine speziellen Anmeldeinformationen angeben. `MONGODB-AWS` Wenn Sie MongoDB-Treiber verwenden, die den `MONGODB-AWS` Authentifizierungsmechanismus unterstützen, können die Treiber auch IAM-Rollenanmeldedaten von der Compute-Instance abrufen (z. B. Amazon EC2, Lambda-Funktion und andere). Im folgenden Beispiel wird eine Mongo-Shell zur Authentifizierung verwendet, `MONGODB-AWS` indem ein Zugriffsschlüssel und ein geheimer Schlüssel (eines IAM-Benutzers) manuell übergeben werden, um die Authentifizierung gegenüber Amazon DocumentDB nachzuweisen.

Im folgenden Beispiel wird Python-Code zur Authentifizierung verwendet, `MONGODB-AWS` ohne dass Anmeldeinformationen explizit übergeben werden (mithilfe einer der Umgebung zugewiesenen IAM-Rolle), um die Authentifizierung gegenüber Amazon DocumentDB zu demonstrieren.

```
##Create a MongoDB client, open a connection to Amazon DocumentDB using an IAM role
    client = pymongo.MongoClient('mongodb://<DocDBEndpoint>:27017/?
tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false&authSource=
%24external&authMechanism=MONGODB-AWS')
```

Im folgenden Beispiel wird eine Mongo-Shell verwendet, um sich mithilfe eines `MONGODB-AWS` Mechanismus zu authentifizieren, indem ein Zugriffsschlüssel und ein geheimer Schlüssel (eines IAM-Benutzers) manuell übergeben werden, um die Authentifizierung gegenüber Amazon DocumentDB nachzuweisen.

```
$ mongo 'mongodb://<access_key>:<secret_key>@<cluster_endpoint>:<db_port>/test?
authSource=%24external&authMechanism=MONGODB-AWS'
```

Im folgenden Beispiel wird eine Mongo-Shell zur Authentifizierung verwendet, `MONGODB-AWS` ohne dass Anmeldeinformationen explizit übergeben werden (mithilfe der der Umgebung zugewiesenen IAM-Rolle), um die Authentifizierung gegenüber Amazon DocumentDB zu demonstrieren.

```
$ mongo 'mongodb://<cluster_endpoint>:<db_port>/test?authSource=
%24external&authMechanism=MONGODB-AWS'
```

Konfiguration von AWS Compute-Typen für die Authentifizierung bei Amazon DocumentDB mithilfe von IAM AWS

Amazon verwenden EC2/AWS Lambda/AWS Fargate

Amazon EC2 verwendet die folgenden Umgebungsvariablen. Wenn der EC2 Instance eine IAM-Rolle oder eine Ausführungs-IAM-Rolle mit einer Lambda-Funktion oder einer Amazon ECS-Aufgabe verknüpft ist, werden diese Variablen automatisch gefüllt und der Treiber kann diese Werte aus der Umgebung abrufen:

```
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY
AWS_SESSION_TOKEN
```

Weitere Informationen zu Umgebungsvariablen finden Sie unter Verwenden von Lambda-Umgebungsvariablen im AWS Lambda Entwicklerhandbuch.

Amazon EKS verwenden

Wenn Sie Ihren Amazon Elastic Kubernetes Service (Amazon EKS) -Pods eine Rolle zuweisen, werden automatisch die folgenden beiden Umgebungsvariablen eingerichtet:

```
AWS_WEB_IDENTITY_TOKEN_FILE - path of web identity token file
AWS_ROLE_ARN - Name of IAM role to connect with
```

Übernehmen Sie mithilfe dieser Variablen manuell die Rolle aus Ihrem Code mithilfe des AWS SDK-Aufrufs für: `AssumeRoleWithWebIdentity`

- Lassen Sie den `ProviderID` Parameter weg.
- Suchen Sie den Wert des `WebIdentityToken` Parameters in der Datei, die in der `AWS_WEB_IDENTITY_TOKEN_FILE` Umgebungsvariablen beschrieben ist.

Weitere Informationen zu Amazon EKS finden Sie unter Was ist Amazon EKS im Amazon EKS-Benutzerhandbuch.

Überwachung von IAM-Authentifizierungsanfragen

Amazon DocumentDB DocumentDB-Auditing verwenden

Gehen Sie zum Audit-Log-Ordner in Amazon und verwenden Sie verschiedene Suchmuster CloudWatch, um die Protokolle für die IAM-Authentifizierung abzurufen. Verwenden Sie es beispielsweise `{ $.param.mechanism = "MONGODB-AWS" }` als Suchmuster für „Alle Protokollstreams durchsuchen“.

Weitere Informationen zu unterstützten Ereignissen bei der Überwachung finden Sie unterAmazon DocumentDB DocumentDB-Ereignisse prüfen.

Verwenden von CloudWatch Amazon-Metriken

`StsGetCallerIdentityCalls`: Diese Metrik zeigt, wie viele `GetCallerIdentity` Aufrufe eine Amazon DocumentDB DocumentDB-Instance an den regionalized AWS Security Token Service (AWS STS) -Endpunkt tätigt. In der `MONGODB-AWS` Authentifizierungsspezifikation erfahren Sie, warum Datenbank-Instances `GetCallerIdentity` STS-Aufrufe tätigen müssen.

Verwenden der IAM-Authentifizierung

Wenn Sie den Benutzernamen und das Passwort nicht in Ihrer eigenen Datenbank verwalten möchten, können Sie die IAM-Authentifizierung verwenden. Die IAM-Authentifizierung ist nur in der instanzbasierten Amazon DocumentDB-Cluster-Version 5.0 verfügbar.

Die IAM-Authentifizierung ist vom STS-Service abhängig. Wir empfehlen Ihnen, zu prüfen, ob Sie die Verbindungsrate senken können, wenn Sie die IAM-Authentifizierung für die Verbindung verwenden und eine STS-Drosselungsausnahme auftritt.

Informationen zu IAM-Kontingenten finden Sie unter IAM und AWS STS Kontingente im IAM-Benutzerhandbuch.

Treiber, die IAM unterstützen

Treiber, die Amazon DocumentDB 5.0 und den `MONGODB-AWS` Authentifizierungsmechanismus unterstützen, sollten mit der IAM-Authentifizierungsimplementierung in Amazon DocumentDB funktionieren.

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js

verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Häufig gestellte Fragen zur IAM-Identitätsauthentifizierung

Gibt es Beispiele, auf die ich mich beziehen kann?

Auf diesen Seiten finden Sie Beispiele für Anwendungsfälle und Konfigurationen:

- So können sich menschliche Benutzer mithilfe von IAM-Benutzern und IAM-Rollen bei Amazon DocumentDB authentifizieren
- Passwortlose Authentifizierung bei Amazon DocumentDB mithilfe von IAM-Rollen

Ich erhalte eine Fehlermeldung bei der Verwendung meines Python-Treibers: „pymongo.errors. ConfigurationError: MONGODB-AWS Authentifizierung erfordert“. pymongo-auth-aws Wie kann ich das lösen?

Bitte stellen Sie sicher, dass Sie bei der Installation des Python-Treibers mit IAM-Authentifizierung die folgende Anweisung verwenden:

```
pip install 'pymongo[aws]'
```

Dadurch werden die zusätzlichen AWS Abhängigkeiten installiert, die für das Funktionieren der IAM-Authentifizierung erforderlich sind.

Wird meine Verbindung unterbrochen, wenn meine temporären Anmeldeinformationen für meine IAM-Rolle ablaufen?

Nein, die temporären IAM-Anmeldeinformationen werden nur für den Verbindungsaufbau und die Authentifizierung verwendet. Dann erfolgt die gesamte weitere Autorisierung im Amazon DocumentDB-Cluster. Selbst wenn die IAM-Anmeldeinformationen rotieren/ablaufen, wird die Verbindung nicht unterbrochen oder veraltet.

Amazon DocumentDB DocumentDB-Benutzer verwalten

In Amazon DocumentDB authentifizieren sich Benutzer bei einem Cluster in Verbindung mit einem Passwort. Jeder Cluster verfügt über primäre Anmeldeinformationen, die bei der Clustererstellung festgelegt werden.

> **Note**
>
> Allen neuen Benutzern, die vor dem 26. März 2020 erstellt wurden, wurden die `dbAdminAnyDatabase`-, `readWriteAnyDatabase`- und `clusterAdmin`-Rollen erteilt. Es wird empfohlen, alle Benutzer neu zu bewerten und die Rollen nach Bedarf zu ändern, um die geringstmöglichen Berechtigungen für alle Benutzer in Ihren Clustern zu erzwingen.
> Weitere Informationen finden Sie unter Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle.

Primär und Benutzer **`serviceadmin`**

Ein neu erstellter Amazon DocumentDB-Cluster hat zwei Benutzer: den Hauptbenutzer und den `serviceadmin` Benutzer.

Der Hauptbenutzer ist ein einzelner, privilegierter Benutzer, der administrative Aufgaben ausführen und zusätzliche Benutzer mit Rollen erstellen kann. Wenn Sie sich zum ersten Mal mit einem Amazon DocumentDB-Cluster verbinden, müssen Sie sich mit den primären Anmeldeinformationen authentifizieren. Der Hauptbenutzer erhält diese Administratorberechtigungen für einen Amazon DocumentDB-Cluster, wenn dieser Cluster erstellt wird, und ihm wird die Rolle von `root` zugewiesen.

Der `serviceadmin`-Benutzer wird implizit beim Erstellen des Clusters erstellt. Jeder Amazon DocumentDB-Cluster hat einen `serviceadmin` Benutzer, AWS der die Möglichkeit bietet, Ihren Cluster zu verwalten. Sie können sich mit `serviceadmin` nicht anmelden, den Benutzer löschen oder umbenennen, sein Passwort ändern oder die Berechtigungen ändern. Jeder Versuch, das zu tun, führt zu einem Fehler.

> **Note**
>
> Die primären `serviceadmin` Benutzer und Benutzer für einen Amazon DocumentDB-Cluster können nicht gelöscht werden, und die Rolle des `root` Hauptbenutzers kann nicht entzogen werden.
> Wenn Sie Ihr primäres Benutzerkennwort vergessen haben, können Sie es mit dem AWS Management Console oder dem zurücksetzen. AWS CLI

Zusätzliche Benutzer erstellen

Nachdem Sie sich als Hauptbenutzer (oder als beliebiger Benutzer mit dieser Rolle`createUser`) verbunden haben, können Sie einen neuen Benutzer erstellen, wie unten gezeigt.

```
db.createUser(
    {
        user: "sample-user-1",
        pwd: "password123",
        roles:
            [{"db":"admin", "role":"dbAdminAnyDatabase" }]
    }
)
```

Um Benutzerdetails anzuzeigen, können Sie den Befehl `show users` wie folgt verwenden. Sie können Benutzer zusätzlich mit dem Befehl `dropUser` entfernen. Weitere Informationen finden Sie unter Allgemeine Befehle.

```
show users
{
    "_id" : "serviceadmin",
    "user" : "serviceadmin",
    "db" : "admin",
    "roles" : [
     {
            "role" : "root",
            "db" : "admin"
        }
    ]
},

{
    "_id" : "myPrimaryUser",
    "user" : "myPrimaryUser",
    "db" : "admin",
    "roles" : [
     {
            "role" : "root",
            "db" : "admin"
        }
    ]
```

```
},

{
    "_id" : "sample-user-1",
    "user" : "sample-user-1",
    "db" : "admin",
    "roles" : [
     {
            "role" : "dbAdminAnyDatabase",
            "db" : "admin"
     }
    ]
}
```

Im obigen Beispiel wird der neue Benutzer `sample-user-1` der `admin`-Datenbank zugewiesen. Dies ist bei einem neuen Benutzer immer der Fall. Amazon DocumentDB hat nicht das Konzept einer `authenticationDatabase` und daher wird die gesamte Authentifizierung im Kontext der `admin` Datenbank durchgeführt.

Wenn Sie beim Erstellen von Benutzern das db Feld bei der Angabe der Rolle weglassen, ordnet Amazon DocumentDB die Rolle implizit der Datenbank zu, für die die Verbindung hergestellt wird. Wenn Ihre Verbindung beispielsweise für die Datenbank `sample-database` hergestellt wird und Sie den folgenden Befehl ausführen, wird der Benutzer `sample-user-2` in der Datenbank `admin` erstellt und verfügt über `readWrite`-Berechtigungen für die Datenbank `sample-database`.

```
db.createUser(
    {
        user: "sample-user-2",
        pwd: "password123",
        roles:
            ["readWrite"]
    }
)
```

Wenn Sie Benutzer mit Rollen erstellen, die über alle Datenbanken hinweg erfasst sind (z. B. `readInAnyDatabase`), müssen Sie sich beim Erstellen des Benutzers entweder im Kontext der `admin`-Datenbank befinden oder beim Erstellen des Benutzers explizit die Datenbank für die Rolle angeben.

Um den Kontext Ihrer Datenbank zu wechseln, können Sie den folgenden Befehl verwenden.

```
use admin
```

Weitere Informationen zur rollenbasierten Zugriffssteuerung und zum Erzwingen geringstmöglicher Berechtigungen unter den Benutzern im Cluster finden Sie unter Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle.

Automatisch rotierende Passwörter für Amazon DocumentDB

Mit AWS Secrets Manager können Sie hartcodierte Anmeldeinformationen in Ihrem Code (einschließlich Kennwörtern) durch einen API-Aufruf an Secrets Manager ersetzen, um das Geheimnis programmgesteuert abzurufen. Dadurch wird sichergestellt, dass das Secret nicht kompromittiert werden kann, wenn jemand Ihren Code durchsucht, da es sich gar nicht dort befindet. Außerdem können Sie Secrets Manager so konfigurieren, dass er das Secret automatisch nach einem von Ihnen festgelegten Zeitplan rotiert. So können Sie Secrets mit langer Einsatzdauer durch Secrets mit kurzer Einsatzdauer ersetzen und damit das Risiko einer Kompromittierung erheblich verringern.

Mit Secrets Manager können Sie Ihre Amazon DocumentDB-Passwörter (d. h. Secrets) mithilfe einer von Secrets Manager AWS Lambda bereitgestellten Funktion automatisch rotieren.

Weitere Informationen AWS Secrets Manager zur systemeigenen Integration mit Amazon DocumentDB finden Sie im Folgenden:

- Blog: So wechseln Sie Amazon DocumentDB- und Amazon Redshift Redshift-Anmeldeinformationen in Secrets Manager AWS
- Was ist AWS Secrets Manager?
- AWS Secrets Manager Geheimnisse rotieren
- Amazon DocumentDB DocumentDB-Anmeldeinformationen in Secrets Manager

Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle

Mithilfe der rollenbasierten Zugriffskontrolle (RBAC) in Amazon DocumentDB (mit MongoDB-Kompatibilität) können Sie den Zugriff auf die Aktionen einschränken, die Benutzer an Datenbanken ausführen können. Bei RBAC werden einem Benutzer eine oder mehrere Rollen gewährt. Diese Rollen bestimmen die Operationen, die ein Benutzer mit Datenbankressourcen ausführen kann.

Amazon DocumentDB unterstützt derzeit sowohl integrierte Rollen, die auf Datenbankebene beschränkt sind, wie z. B.,,,,,,`read`,,,`readWrite`,, `readAnyDatabase``clusterAdmin`, als auch benutzerdefinierte Rollen, die auf bestimmte Aktionen beschränkt werden können, und granulare Ressourcen wie Sammlungen, die auf Ihre Anforderungen zugeschnitten sind.

Zu den häufigsten Anwendungsfällen für RBAC gehören die Durchsetzung der geringsten Rechte durch die Erstellung von Benutzern mit schreibgeschütztem Zugriff auf die Datenbanken oder Sammlungen in einem Cluster sowie mandantenfähige Anwendungsdesigns, die es einem einzelnen Benutzer ermöglichen, auf eine bestimmte Datenbank oder Sammlung in einem Cluster zuzugreifen.

> **Note**
>
> Allen neuen Benutzern, die vor dem 26. März 2020 erstellt wurden, wurden die `dbAdminAnyDatabase`-, `readWriteAnyDatabase`- und `clusterAdmin`-Rollen erteilt. Es wird empfohlen, alle vorhandenen Benutzer neu zu bewerten und die Rollen nach Bedarf zu ändern, um die geringstmöglichen Berechtigungen für Ihre Cluster zu erzwingen.

Themen

- RBAC-Konzepte
- Erste Schritte mit den integrierten RBAC-Rollen
- Erste Schritte mit benutzerdefinierten RBAC-Rollen
- Als Benutzer eine Verbindung zu Amazon DocumentDB herstellen
- Allgemeine Befehle
- Funktionelle Unterschiede
- Einschränkungen
- Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle

RBAC-Konzepte

Im Folgenden finden Sie wichtige Begriffe und Konzepte zur rollenbasierten Zugriffssteuerung. Weitere Informationen zu Amazon DocumentDB DocumentDB-Benutzern finden Sie unterAmazon DocumentDB DocumentDB-Benutzer verwalten.

- Benutzer — Eine einzelne Entität, die sich bei der Datenbank authentifizieren und Operationen ausführen kann.

- Passwort — Ein Geheimnis, das zur Authentifizierung des Benutzers verwendet wird.
- Rolle — Autorisiert einen Benutzer, Aktionen an einer oder mehreren Datenbanken durchzuführen.
- Admin-Datenbank — Die Datenbank, in der Benutzer gespeichert sind und für die sie autorisiert sind.
- Datenbank (**db**) — Der Namespace innerhalb von Clustern, der Sammlungen zum Speichern von Dokumenten enthält.

Der folgende Befehl erstellt einen Benutzer mit Namen `sample-user`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: [{role: "read", db: "sample-database"}]})
```

In diesem Beispiel:

- `user: "sample-user"`— Gibt den Benutzernamen an.
- `pwd: "abc123"`— Zeigt das Benutzerkennwort an.
- `role: "read", "db: "sample-database"`— Zeigt an, dass der Benutzer Leseberechtigungen für haben `sample-user` wird`sample-database`.



Das folgende Beispiel zeigt die Ausgabe, nachdem Sie den Benutzer `sample-user` mit `db.getUser(sample-user)` erhalten. In diesem Beispiel befindet sich der Benutzer `sample-user` in der Datenbank `admin`, besitzt jedoch die „read“-Rolle für die Datenbank `sample-database`.



Wenn Sie beim Erstellen von Benutzern das db Feld bei der Angabe der Rolle weglassen, ordnet Amazon DocumentDB die Rolle implizit der Datenbank zu, für die die Verbindung hergestellt wird. Wenn Ihre Verbindung beispielsweise für die Datenbank `sample-database` hergestellt wird und Sie den folgenden Befehl ausführen, wird der Benutzer `sample-user` in der Datenbank `admin` erstellt und verfügt über `readWrite`-Berechtigungen für die Datenbank `sample-database`.

```
db.createUser({user: "sample-user", pwd: "abc123", roles: ["readWrite"]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"sample-user",
   "roles":[
      {
         "db":"sample-database",
         "role":"readWrite"
      }
   ]
}
```

Wenn Sie Benutzer mit Rollen erstellen, die über alle Datenbanken hinweg erfasst sind (z. B. `readAnyDatabase`), müssen Sie sich beim Erstellen des Benutzers entweder im Kontext der Datenbank `admin` befinden oder beim Erstellen des Benutzers explizit die Datenbank für die Rolle angeben. Um Befehle für die Datenbank `admin` auszugeben, können Sie den Befehl `use admin` verwenden. Weitere Informationen finden Sie unter Allgemeine Befehle.

Erste Schritte mit den integrierten RBAC-Rollen

Um Ihnen den Einstieg in die rollenbasierte Zugriffssteuerung zu erleichtern, führen Sie in diesem Abschnitt ein Beispielszenario zum Erzwingen der geringstmöglichen Berechtigungen durch das Erstellen von Rollen für drei Benutzer mit unterschiedlichen Auftragsfunktionen durch.

- `user1` ist ein neuer Manager, der in der Lage sein muss, alle Datenbanken in einem Cluster anzuzeigen und darauf zuzugreifen.
- `user2` ist ein neuer Mitarbeiter, der Zugriff auf nur eine Datenbank, `sample-database-1`, in demselben Cluster benötigt.
- `user3` ist ein vorhandener Mitarbeiter, der im selben Cluster eine andere Datenbank, `sample-database-2`, anzeigen und darauf zugreifen muss, auf die er vorher keinen Zugriff hatten.

Zu einem späteren Zeitpunkt verlassen sowohl `user1` als auch `user2` das Unternehmen, so dass ihr Zugang widerrufen werden muss.

Um Benutzer zu erstellen und Rollen zu erteilen, muss der Benutzer, mit dem Sie sich beim Cluster authentifizieren, über eine zugeordnete Rolle verfügen, die Aktionen für `createUser` und `grantRole` ausführen kann. Zum Beispiel können die Rollen `admin` und `userAdminAnyDatabase` beide solche Fähigkeiten gewähren. Informationen zu Aktionen pro Rolle finden Sie unter Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle.

> **Note**
>
> In Amazon DocumentDB werden alle Benutzer- und Rollenoperationen (z. B.`create`,`get`,`drop`, `grantrevoke`, usw.) implizit in der `admin` Datenbank ausgeführt, unabhängig davon, ob Sie Befehle für die `admin` Datenbank ausgeben oder nicht.

Um zunächst zu verstehen, was die aktuellen Benutzer und Rollen im Cluster sind, können Sie den Befehl `show users` ausführen, wie im folgenden Beispiel. Sie sehen zwei Benutzer `serviceadmin` und den Hauptbenutzer für den Cluster. Diese beiden Benutzer sind immer vorhanden und können nicht gelöscht werden. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Benutzer verwalten.

```
show users
```

Erstellen Sie für `user1` eine Rolle mit Lese- und Schreibzugriff auf alle Datenbanken im gesamten Cluster mit dem folgenden Befehl.

```
db.createUser({user: "user1", pwd: "abc123", roles: [{role: "readWriteAnyDatabase", db:
 "admin"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Erstellen Sie für `user2` eine Rolle mit schreibgeschütztem Zugriff auf die Datenbank `sample-database-1` mit dem folgenden Befehl.

```
db.createUser({user: "user2", pwd: "abc123", roles: [{role: "read", db: "sample-
database-1"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user2",
   "roles":[
      {
         "role":"read",
         "db":"sample-database-1"
      }
   ]
}
```

Um das Szenario zu simulieren, dass `user3` ein vorhandener Benutzer ist, erstellen Sie zuerst den Benutzer `user3` und weisen Sie dann `user3` eine neue Rolle zu.

```
db.createUser({user: "user3", pwd: "abc123", roles: [{role: "readWrite", db: "sample-
database-1"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user3",
   "roles":[
      {
         "role":"readWrite",
         "db":"sample-database-1"
      }
   ]
}
```

Nachdem der Benutzer `user3` erstellt wurde, weisen Sie `user3` die `read`-Rolle für `sample-database-2` zu.

```
db.grantRolesToUser("user3", [{role: "read", db: "sample-database-2"}])
```

Schließlich verlassen sowohl `user1` als auch `user2` das Unternehmen und ihr Zugriff auf den Cluster muss widerrufen werden. Sie können dies tun, indem Sie die Benutzer wie folgt löschen.

```
db.dropUser("user1")
db.dropUser("user2")
```

Um sicherzustellen, dass alle Benutzer über die entsprechenden Rollen verfügen, können Sie alle Benutzer mit dem folgenden Befehl auflisten.

```
show users
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "_id":"serviceadmin",
   "user":"serviceadmin",
   "db":"admin",
   "roles":[
```

```
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"master-user",
   "user":"master-user",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"user3",
   "user":"user3",
   "db":"admin",
   "roles":[
      {
         "db":"sample-database-2",
         "role":"read"
      },
      {
         "db":"sample-database-1",
         "role":"readWrite"
      }
   ]
}
```

Erste Schritte mit benutzerdefinierten RBAC-Rollen

Um Ihnen den Einstieg in benutzerdefinierte Rollen zu erleichtern, führt Sie dieser Abschnitt durch ein Beispielszenario zur Durchsetzung der geringsten Rechte durch die Erstellung von Rollen für drei Benutzer mit unterschiedlichen Aufgabenfunktionen.

In diesem Beispiel gilt Folgendes:

- `user1` ist ein neuer Manager, der in der Lage sein muss, alle Datenbanken in einem Cluster anzuzeigen und darauf zuzugreifen.

- `user2`ist ein neuer Mitarbeiter, der nur die Aktion „Suchen“ für nur eine Datenbank im selben Cluster benötigt. `sample-database-1`
- `user3`ist ein vorhandener Mitarbeiter, der eine bestimmte Sammlung (Spalte 2) in einer anderen Datenbank, auf `sample-database-2` die er zuvor keinen Zugriff hatte, im selben Cluster anzeigen und darauf zugreifen muss.
- Erstellen Sie für `user1` eine Rolle mit Lese- und Schreibzugriff auf alle Datenbanken im gesamten Cluster mit dem folgenden Befehl.

```
db.createUser(
 {
     user: "user1", pwd: "abc123",
     roles: [{role: "readWriteAnyDatabase", db: "admin"}]
 }
 )
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"user1",
   "roles":[
      {
         "role":"readWriteAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Erstellen Sie für `user2` mit dem folgenden Befehl eine Rolle mit Find-Rechten für alle Sammlungen in der Datenbank`sample-database-1`. Beachten Sie, dass diese Rolle sicherstellen würde, dass alle zugehörigen Benutzer nur Suchabfragen ausführen können.

```
db.createRole(
{
    role: "findRole",
    privileges: [
    {
        resource: {db: "sample-database-1", collection: ""}, actions: ["find"]
    }],
    roles: []
```

```
}
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"findRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "find"
         ]
      }
   ],
   "roles":[

   ]
}
```

Erstellen Sie als Nächstes den Benutzer (`user2`) und fügen Sie dem Benutzer die kürzlich erstellte Rolle `findRole` hinzu.

```
db.createUser(
{
    user: "user2",
    pwd: "abc123",
    roles: []
})

 db.grantRolesToUser("user2",["findRole"])
```

Um das `user3` Szenario eines vorhandenen Benutzers zu simulieren, erstellen Sie zuerst den Benutzer `user3` und dann eine neue Rolle namens CollectionRole, der wir im nächsten Schritt zuweisen werden. `user3`

Jetzt können Sie eine neue Rolle zuweisen. `user3` Diese neue Rolle ermöglicht es `user3` Ihnen, eine bestimmte Sammlung einzufügen, zu aktualisieren, zu löschen und Zugriff auf diese zu finden. `sample-database-2`

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Nachdem der Benutzer erstellt `user3` wurde, können Sie ihm `user3` die Rolle `collectionFind` zuweisen.

```
db.grantRolesToUser("user3",["collectionRole"])
```

Schließlich verlassen sowohl `user1` als auch `user2` das Unternehmen und ihr Zugriff auf den Cluster muss widerrufen werden. Sie können dies tun, indem Sie die Benutzer wie folgt löschen.

```
db.dropUser("user1")
db.dropUser("user2")
```

Um sicherzustellen, dass alle Benutzer über die entsprechenden Rollen verfügen, können Sie alle Benutzer mit dem folgenden Befehl auflisten.

```
show users
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "_id":"serviceadmin",
   "user":"serviceadmin",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
   "_id":"master-user",
   "user":"master-user",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"root"
      }
   ]
}
{
```

```
   "_id":"user3",
   "user":"user3",
   "db":"admin",
   "roles":[
      {
         "db":"admin",
         "role":"collectionRole"
      }
   ]
}
```

Als Benutzer eine Verbindung zu Amazon DocumentDB herstellen

Wenn Sie eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen, stellen Sie eine Verbindung im Kontext einer bestimmten Datenbank her. Wenn Sie in der Verbindungszeichenfolge keine Datenbank angeben, werden Sie standardmäßig automatisch im Kontext der Datenbank `test` mit dem Cluster verbunden. Alle Befehle auf Sammlungsebene wie `insert` und `find` werden für Sammlungen in der Datenbank `test` ausgegeben.

Um die Datenbank zu sehen, in der Sie sich befinden, oder — mit anderen Worten — Befehle für die Sie ausführen, verwenden Sie den db Befehl in der Mongo-Shell wie folgt.

Abfrage:

```
db
```

Ausgabe:

```
test
```

Obwohl sich die Standardverbindung möglicherweise im Kontext der Datenbank `test` befindet, bedeutet dies nicht unbedingt, dass der Benutzer, der der Verbindung zugeordnet ist, berechtigt ist, Aktionen für die Datenbank `test` auszuführen. Wenn Sie sich im vorangegangenen Beispielszenario als Benutzer `user3` authentifizieren, der die `readWrite`-Rolle für die Datenbank `sample-database-1` besitzt, ist der Standardkontext Ihrer Verbindung die Datenbank `test`. Wenn Sie jedoch versuchen, ein Dokument in eine Sammlung in der Datenbank `test` einzufügen, erhalten Sie die Fehlermeldung Autorisierungsfehler. Dies liegt daran, dass dieser Benutzer nicht berechtigt ist, diesen Befehl in dieser Datenbank auszuführen, wie unten gezeigt.

Abfrage:

```
db
```

Ausgabe:

```
test
```

Abfrage:

```
db.col.insert({x:1})
```

Ausgabe:

```
WriteCommandError({ "ok" : 0, "code" : 13, "errmsg" : "Authorization failure" })
```

Wenn Sie den Kontext Ihrer Verbindung zur Datenbank `sample-database-1` ändern, können Sie in die Sammlung schreiben, für die der Benutzer die Berechtigung dazu hat.

Abfrage:

```
use sample-database-1
```

Ausgabe:

```
switched to db sample-database-1
```

Abfrage:

```
db.col.insert({x:1})
```

Ausgabe:

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

Wenn Sie sich bei einem Cluster mit einem bestimmten Benutzer authentifizieren, können Sie die Datenbank auch in der Verbindungszeichenfolge angeben. Dadurch entfällt die Notwendigkeit, den Befehl `use` auszuführen, nachdem der Benutzer in der Datenbank `admin` authentifiziert wurde.

Die folgende Verbindungszeichenfolge authentifiziert den Benutzer für die Datenbank `admin`, aber der Kontext der Verbindung wird für die Datenbank `sample-database-1` verwendet.

```
mongo "mongodb://user3:abc123@sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
sample-database-2"
```

Allgemeine Befehle

Dieser Abschnitt enthält Beispiele für häufig verwendete Befehle, die die rollenbasierte Zugriffskontrolle in Amazon DocumentDB verwenden. Sie müssen sich im Kontext der Datenbank `admin` befinden, um Benutzer und Rollen zu erstellen und zu ändern. Sie können den Befehl `use admin` verwenden, um zur Datenbank `admin` zu wechseln.

> **Note**
>
> Änderungen an den Benutzern und Rollen erfolgen implizit in der Datenbank `admin`. Wenn Sie Benutzer mit Rollen erstellen, die über alle Datenbanken hinweg erfasst sind (z. B. `readAnyDatabase`), müssen Sie sich beim Erstellen des Benutzers entweder im Kontext der Datenbank `admin` befinden (d. h. `use admin`) oder beim Erstellen des Benutzers explizit die Datenbank für die Rolle angeben (wie in Beispiel 2 in diesem Abschnitt gezeigt).

Beispiel 1: Erstellen Sie einen Benutzer mit `read` Rolle für die Datenbank. `foo`

```
db.createUser({user: "readInFooBar", pwd: "abc123", roles: [{role: "read", db:
 "foo"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"readInFooBar",
   "roles":[
      {
         "role":"read",
         "db":"foo"
      }
   ]
}
```

Beispiel 2: Erstellen Sie einen Benutzer mit Lesezugriff auf alle Datenbanken.

```
db.createUser({user: "readAllDBs", pwd: "abc123", roles: [{role: "readAnyDatabase", db:
 "admin"}]})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "user":"readAllDBs",
   "roles":[
      {
         "role":"readAnyDatabase",
         "db":"admin"
      }
   ]
}
```

Beispiel 3: Erteilen Sie einem vorhandenen Benutzer eine `read` Rolle in einer neuen Datenbank.

```
db.grantRolesToUser("readInFooBar", [{role: "read", db: "bar"}])
```

Beispiel 4: Aktualisieren Sie die Rolle eines Benutzers.

```
db.updateUser("readInFooBar", {roles: [{role: "read", db: "foo"}, {role: "read", db:
 "baz"}]})
```

Beispiel 5: Widerrufen Sie einem Benutzer den Zugriff auf eine Datenbank.

```
db.revokeRolesFromUser("readInFooBar", [{role: "read", db: "baz"}])
```

Beispiel 6: Beschreiben Sie eine integrierte Rolle.

```
db.getRole("read", {showPrivileges:true})
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"read",
   "db":"sample-database-1",
   "isBuiltin":true,
   "roles":[
```

```
   ],
   "inheritedRoles":[

   ],
   "privileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "changeStream",
            "collStats",
            "dbStats",
            "find",
            "killCursors",
            "listCollections",
            "listIndexes"
         ]
      }
   ],
   "inheritedPrivileges":[
      {
         "resource":{
            "db":"sample-database-1",
            "collection":""
         },
         "actions":[
            "changeStream",
            "collStats",
            "dbStats",
            "find",
            "killCursors",
            "listCollections",
            "listIndexes"
         ]
      }
   }
```

Beispiel 7: Einen Benutzer aus dem Cluster löschen.

```
db.dropUser("readInFooBar")
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
true
```

Beispiel 8: Erstellen Sie eine Rolle mit Lese- und Schreibzugriff auf eine bestimmte Sammlung

```
db.createRole(
{
    role: "collectionRole",
    privileges: [
    {
        resource: {db: "sample-database-2", collection: "col2"}, actions: ["find",
 "update", "insert", "remove"]
    }],
    roles: []
}
)
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
   "role":"collectionRole",
   "privileges":[
      {
         "resource":{
            "db":"sample-database-2",
            "collection":"col2"
         },
         "actions":[
            "find",
            "update",
            "insert",
            "remove"
         ]
      }
   ],
   "roles":[

   ]
}
```

Beispiel 9: Erstellen Sie einen Benutzer und weisen Sie ihm eine benutzerdefinierte Rolle zu

```
db.createUser(
{
    user: "user3",
    pwd: "abc123",
    roles: []
})

db.grantRolesToUser("user3",["collectionRole"])
```

Beispiel 10: Erteilen Sie einer benutzerdefinierten Rolle zusätzliche Rechte

```
db.grantPrivilegesToRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col1" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Beispiel 11: Entfernen Sie Rechte aus einer benutzerdefinierten Rolle

```
db.revokePrivilegesFromRole(
  "collectionRole",
  [
    {
      resource: { db: "sample-database-1", collection: "col2" },
      actions: ["find", "update", "insert", "remove"]
    }
  ]
)
```

Beispiel 12: Aktualisieren Sie eine bestehende benutzerdefinierte Rolle

```
db.updateRole(
  "collectionRole",
  {
    privileges: [
    {
        resource: {db: "sample-database-3", collection: "sample-collection-3"},
 actions: ["find", "update", "insert", "remove"]
```

```
    }],
    roles: []
  }
)
```

Funktionelle Unterschiede

In Amazon DocumentDB werden Benutzer- und Rollendefinitionen in der `admin` Datenbank gespeichert und Benutzer werden anhand der `admin` Datenbank authentifiziert. Diese Funktionalität unterscheidet sich von der MongoDB Community Edition, ist aber konsistent mit MongoDB Atlas.

Amazon DocumentDB unterstützt auch Change-Streams, die eine zeitlich geordnete Abfolge von Änderungsereignissen bereitstellen, die in den Sammlungen Ihres Clusters auftreten. Die `listChangeStreams` Aktion wird auf Clusterebene (d. h. auf alle Datenbanken) angewendet, und die `modifyChangeStreams` Aktion kann auf Datenbank- und Clusterebene angewendet werden.

Einschränkungen

Die folgende Tabelle enthält die Grenzwerte für die rollenbasierte Zugriffskontrolle in Amazon DocumentDB.

Beschreibung	Limit
Anzahl Benutzer pro Cluster	1000
Anzahl der Rollen, die einem Benutzer zugeordnet sind	1000
Anzahl der benutzerdefinierten Rollen	100
Anzahl der Ressourcen, die mit einer Berechtigung verknüpft sind	100

Datenbankzugriff mithilfe der rollenbasierten Zugriffskontrolle

Mit der rollenbasierten Zugriffssteuerung können Sie einen Benutzer erstellen und ihm eine oder mehrere Rollen erteilen, um zu bestimmen, welche Operationen der Benutzer in einer Datenbank oder einem Cluster ausführen kann.

Im Folgenden finden Sie eine Liste der integrierten Rollen, die derzeit in Amazon DocumentDB unterstützt werden.

> **Note**
>
> In Amazon DocumentDB 4.0 und 5.0 können die `ListDatabase` Befehle `ListCollection` und optional die `authorizedDatabases` Parameter `authorizedCollections` und verwenden, um die Sammlungen und Datenbanken aufzulisten, auf die der Benutzer zugreifen darf, wobei er die jeweiligen `listDatabase` Rollen `listCollections` und benötigt. Außerdem haben Benutzer jetzt die Möglichkeit, ihre eigenen Cursor zu beenden, ohne die `KillCursor` Rolle zu benötigen.

Database user

Rollenname	Beschreibung	Aktionen

`read`	Gewährt einem Benutzer Lesezugriff auf die angegebene Datenbank.	`changeStreams`
		`collStats`
		`dbStats`
		`find`
		`killCursors`
		`listIndexes`
		`listCollections`

`readWrite`	Gewährt dem Benutzer Lese- und Schreibzugriff auf die angegebene Datenbank.	Alle Aktionen von `read`-Berechtigungen.
		`createCollection`
		`dropCollection`

Rollenname	Beschreibung	Aktionen

		`createIndex`
		`dropIndex`
		`insert`
		`killCursors`
		`listIndexes`
		`listCollections`
		`remove`
		`update`

Cluster user

Rollenname	Beschreibung	Aktionen

readAnyDatabase	Gewährt einem Benutzer Lesezugriff auf alle Datenbanken im Cluster.	Alle Aktionen von `read`-Berechtigungen.
		`listChangeStreams`
		`listDatabases`

`readWriteAnyDatabase`	Gewährt einem Benutzer Lese- und Schreibzugriff auf alle Datenbanken im Cluster.	Alle Aktionen von `readWrite` -Berechtigungen.
		`listChangeStreams`
		`listDatabases`

Rollenname	Beschreibung	Aktionen

`userAdminAnyDatabase`	Gewährt einem Benutzer die Möglichkeit, die Rollen oder Berechtigungen, die ein Benutzer hat, der angegebenen Datenbank zuzuweisen und zu ändern.	`changeCustomData`
		`changePassword`
		`createUser`
		`dropRole`
		`dropUser`
		`grantRole`
		`listDatabases`
		`revokeRole`
		`viewRole`
		`viewUser`

`dbAdminAnyDatabase`	Gewährt einem Benutzer die Möglichkeit, Datenbankverwaltungsrollen für jede angegebene Datenbank auszuführen.	Alle Aktionen von `dbAdmin`-Berechtigungen.
		`dropCollection`
		`listDatabases`
		`listChangeStreams`
		`modifyChangeStreams`

Superuser

Rollenname	Beschreibung	Aktionen
`root`	Gewährt einem Benutzer Zugriff auf die Ressourcen und Operationen aller folgenden Rollen zusammen: `readWriteAnyDatabase` , `dbAdminAnyDatabase` , `userAdminAnyDatabase` , `clusterAdmin` , `restore` und `backup`.	Alle Aktionen von `readWriteAnyDatabase` , `dbAdminAnyDatabase` , `userAdminAnyDatabase` , `clusterAdmin` , `restore` und `backup`.

Database administrator

Rollenname	Beschreibung	Aktionen

`dbAdmin`	Gewährt einem Benutzer die Möglichkeit, administrative Aufgaben für die angegebene Datenbank auszuführen.	`bypassDocumentValidation`
		`collMod`
		`collStats`
		`createCollection`
		`createIndex`
		`dropCollection`
		`dropDatabase`
		`dropIndex`
		`dbStats`

Rollenname	Beschreibung	Aktionen

		`find`
		`killCursors`
		`listIndexes`
		`listCollections`
		`modifyChangeStreams`

`dbOwner`	Gewährt einem Benutzer die Möglichkeit, alle administrativen Aufgaben für die angegebenen Datenbank auszuführen, indem die Rollen `dbAdmin` und `readWrite` kombiniert werden.	Alle Aktionen von `dbAdmin` und `readWrite` .

Cluster administrator

Name der Rolle	Beschreibung	Aktionen

`clusterAdmin`	Gewährt einem Benutzer den größtmöglichen Zugriff auf die Cluster-Verwaltung durch Kombination der Rollen `clusterManager` , `clusterMonitor` und `hostManager` .	Alle Aktionen von `clusterManager` , `clusterMonitor` und `hostManager` .
		`listChangeStreams`
		`dropDatabase`
		`modifyChangeStreams`

Name der Rolle	Beschreibung	Aktionen

`clusterManager`	Gewährt einem Benutzer die Möglichkeit, Verwaltungs- und Überwachungsaktionen für den angegebenen Cluster durchzuführen.	`listChangeStreams`
		`listSessions`
		`modifyChangeStreams`
		`replSetGetConfig`

Name der Rolle	Beschreibung	Aktionen

`clusterMonitor`	Gewährt einem Benutzer die Möglichkeit, schreibgeschützten Zugriff auf Überwachungstools zu haben.	`collStats`
		`dbStats`
		`find`
		`getParameter`
		`hostInfo`
		`indexStats`
		`killCursors`
		`listChangeStreams`
		`listCollections`
		`listDatabases`
		`listIndexes`
		`listSessions`
		`replSetGetConfig`
		`serverStatus`
		`top`

Name der Rolle	Beschreibung	Aktionen

`hostManager`	Gewährt einem Benutzer die Möglichkeit, Server zu überwachen und zu verwalten.	`auditConfigure`
		`killCursors`
		`killAnyCursor`
		`killAnySession`
		`killop`

Backup administrator

Rollenname	Beschreibung	Aktionen

`backup`	Gewährt einem Benutzer den erforderlichen Zugriff zum Sichern von Daten.	`getParameter`
		`insert`
		`find`
		`listChangeStreams`
		`listCollections`
		`listDatabases`
		`listIndexes`
		`update`

`restore`	Gewährt einem Benutzer den erforderlichen Zugriff zum Wiederherstellen von Daten.	`bypassDocumentValidation` `changeCustomData`

Rollenname	Beschreibung	Aktionen
		changePassword collMod createCollection createIndex createUser dropCollection dropRole dropUser getParameter grantRole find insert listCollections modifyChangeStreams revokeRole remove viewRole viewUser update

Protokollierung und Überwachung in Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) bietet eine Vielzahl von CloudWatch Amazon-Metriken, die Sie überwachen können, um den Zustand und die Leistung Ihrer Amazon DocumentDB-Cluster und -Instances zu ermitteln. Sie können Amazon DocumentDB-Metriken mit verschiedenen Tools anzeigen, darunter der Amazon DocumentDB DocumentDB-Konsole AWS CLI, der CloudWatch Amazon-Konsole und der CloudWatch API. Weitere Informationen zur Überwachung finden Sie unter Überwachung von Amazon DocumentDB.

Zusätzlich zu den CloudWatch Amazon-Metriken können Sie den Profiler verwenden, um die Ausführungszeit und Details der Operationen zu protokollieren, die auf Ihrem Cluster ausgeführt wurden. Profiler ist nützlich für die Überwachung der langsamsten Operationen in Ihrem Cluster, um die Leistung einzelner Abfragen und die allgemeine Cluster-Leistung zu verbessern. Wenn diese Option aktiviert ist, werden Vorgänge in Amazon CloudWatch Logs protokolliert, und Sie können CloudWatch Insight verwenden, um Ihre Amazon DocumentDB DocumentDB-Profilerstellungsdaten zu analysieren, zu überwachen und zu archivieren. Weitere Informationen finden Sie unter Profilierung von Amazon DocumentDB DocumentDB-Vorgängen.

Amazon DocumentDB ist auch in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen von Benutzern, Rollen oder einem AWS Service in Amazon DocumentDB (mit MongoDB-Kompatibilität) bereitstellt. CloudTrail erfasst alle AWS CLI API-Aufrufe für Amazon DocumentDB als Ereignisse, einschließlich Aufrufe von Amazon DocumentDB AWS Management Console und von Codeaufrufen an das Amazon DocumentDB-SDK. Weitere Informationen finden Sie unter Protokollieren Amazon DocumentDB DocumentDB-API-Aufrufen mit AWS CloudTrail.

Mit Amazon DocumentDB können Sie Ereignisse prüfen, die in Ihrem Cluster durchgeführt wurden. Beispiele für protokollierte Ereignisse sind erfolgreiche und fehlgeschlagene Authentifizierungsversuche, Drop-Ereignisse für Sammlungen in einer Datenbank oder das Erstellen eines Index. Standardmäßig ist die Prüfung in Amazon DocumentDB deaktiviert und erfordert, dass Sie sich für diese Funktion anmelden. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.

Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate

Themen

- Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters
- Automatische Rotation von Serverzertifikaten

- Fehlerbehebung
- Häufig gestellte Fragen

Das Zertifikat der Zertifizierungsstelle (CA) für Amazon DocumentDB-Cluster wurde im August 2024 aktualisiert. Wenn Sie Amazon DocumentDB-Cluster mit aktiviertem Transport Layer Security (TLS) verwenden (Standardeinstellung) und Sie Ihre Client-Anwendungs- und Serverzertifikate nicht rotiert haben, sind die folgenden Schritte erforderlich, um Verbindungsprobleme zwischen Ihrer Anwendung und Ihren Amazon DocumentDB-Clustern zu beheben.

- Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung
- Schritt 2: Aktualisieren des Serverzertifikats

Die CA- und Serverzertifikate wurden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Client-Anwendungen müssen die neuen CA-Zertifikate zu ihren Trust Stores hinzufügen, und bestehende Amazon DocumentDB DocumentDB-Instances müssen aktualisiert werden, sodass sie die neuen CA-Zertifikate vor diesem Ablaufdatum verwenden können.

Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters

Führen Sie die Schritte in diesem Abschnitt aus, um das CA-Zertifikatspaket Ihrer Anwendung (Schritt 1) und die Serverzertifikate Ihres Clusters (Schritt 2) zu aktualisieren. Bevor Sie die Änderungen auf Ihre Produktionsumgebungen anwenden, empfehlen wir dringend, diese Schritte in einer Entwicklungs- oder Stagingumgebung zu testen.

> **Note**
>
> Sie müssen die Schritte 1 und 2 jeweils AWS-Region ausführen, wenn Sie Amazon DocumentDB-Cluster haben.

Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung

Laden Sie das neue CA-Zertifikat herunter und aktualisieren Sie Ihre Anwendung so, dass sie das neue CA-Zertifikat verwendet, um TLS-Verbindungen zu Amazon DocumentDB herzustellen. Laden Sie das neue CA-Zertifikatpaket von https://truststore.pki.rds.amazonaws.com/global/global-

bundle.pem herunter. Diese Operation lädt die Datei mit dem Namen `global-bundle.pem` herunter.

> **Note**
>
> Wenn Sie auf den Keystore zugreifen, der sowohl das alte CA-Zertifikat (`rds-ca-2019-root.pem`) als auch die neuen CA-Zertifikate (`rds-ca-rsa2048-g1`,,) enthält, stellen Sie sicher`rds-ca-rsa4096-g1`, `rds-ca-ecc384-g1` dass der Keystore auswählt. `global-bundle`

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Aktualisieren Sie als Nächstes Ihre Anwendungen, um das neue Zertifikatpaket zu verwenden. Das neue CA-Paket enthält sowohl das alte CA-Zertifikat (rds-ca-2019) als auch die neuen CA-Zertifikate (2048-g1, 4096-g1, 384-g1). rds-ca-rsa rds-ca-rsa rds-ca-ecc Wenn beide CA-Zertifikate im neuen CA-Paket enthalten sind, können Sie Ihre Anwendung und Ihren Cluster in zwei Schritten aktualisieren.

Für Java-Anwendungen müssen Sie einen neuen Trust Store mit dem neuen CA-Zertifikat erstellen. Anweisungen finden Sie auf der Registerkarte Java im Verbindung bei aktiviertem TLS herstellen Thema.

Informationen dazu, wie Sie überprüfen, ob Ihre Anwendung das neueste CA-Zertifikatspaket verwendet, finden Sie unter Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?. Wenn Sie das neueste CA-Zertifikatspaket in Ihrer Anwendung bereits verwenden, können Sie mit Schritt 2 fortfahren.

Beispiele für die Verwendung eines Zertifizierungsstellenpakets mit Ihrer Anwendung finden Sie unter Verschlüsseln von Daten während der Übertragung und Verbindung bei aktiviertem TLS herstellen.

> **Note**
>
> Zurzeit akzeptiert der MongoDB Go Driver 1.2.1 nur ein einzelnes CA-Serverzertifikat in `sslcertificateauthorityfile`. Informationen dazu, wie Sie bei aktiviertem TLS eine Verbindung zu Amazon DocumentDB über Go herstellen, finden Sie unter Verbindung bei aktiviertem TLS herstellen.

Schritt 2: Aktualisieren des Serverzertifikats

Nachdem die Anwendung für die Verwendung des neuen CA-Bundles aktualisiert wurde, besteht der nächste Schritt darin, das Serverzertifikat zu aktualisieren, indem jede Instance in einem Amazon DocumentDB-Cluster geändert wird. Informationen zum Ändern von Instances zur Verwendung des neuen Serverzertifikats finden Sie in den folgenden Anweisungen.

Amazon DocumentDB bietet Folgendes CAs , um das DB-Serverzertifikat für eine DB-Instance zu signieren:

- rds-ca-ecc384-g1 — Verwendet eine Zertifizierungsstelle mit ECC 384-Algorithmus für private Schlüssel und Signaturalgorithmus. SHA384 Diese CA unterstützt die automatische Rotation von Serverzertifikaten. Dies wird nur auf Amazon DocumentDB 4.0 und 5.0 unterstützt.
- rds-ca-rsa2048-g1 — Verwendet in den meisten Regionen eine Zertifizierungsstelle mit dem RSA 2048-Algorithmus für private Schlüssel und dem Signaturalgorithmus. SHA256 AWS Diese CA unterstützt die automatische Rotation von Serverzertifikaten.
- rds-ca-rsa4096-g1 — Verwendet eine Zertifizierungsstelle mit dem RSA 4096-Algorithmus für private Schlüssel und dem Signaturalgorithmus. SHA384 Diese CA unterstützt die automatische Rotation von Serverzertifikaten.

> **Note**
>
> Wenn Sie den verwenden AWS CLI, können Sie die Gültigkeiten der oben aufgeführten Zertifizierungsstellen mithilfe von Describe-Zertifikaten überprüfen.

Diese CA-Zertifikate sind im regionalen und globalen Zertifikat-Bundle enthalten. Wenn Sie die CA rds-ca-rsa 2048-g1, rds-ca-rsa 4096-g1 oder rds-ca-ecc 384-g1 mit einer Datenbank verwenden, verwaltet Amazon DocumentDB das DB-Serverzertifikat in der Datenbank. Amazon DocumentDB rotiert das DB-Serverzertifikat automatisch, bevor es abläuft.

> **Note**
>
> Amazon DocumentDB erfordert keinen Neustart für die Zertifikatsrotation, wenn Ihr Cluster auf den folgenden Engine-Patch-Versionen läuft:
>
> - Amazon DocumentDB 3.6:1.0.208662 oder höher

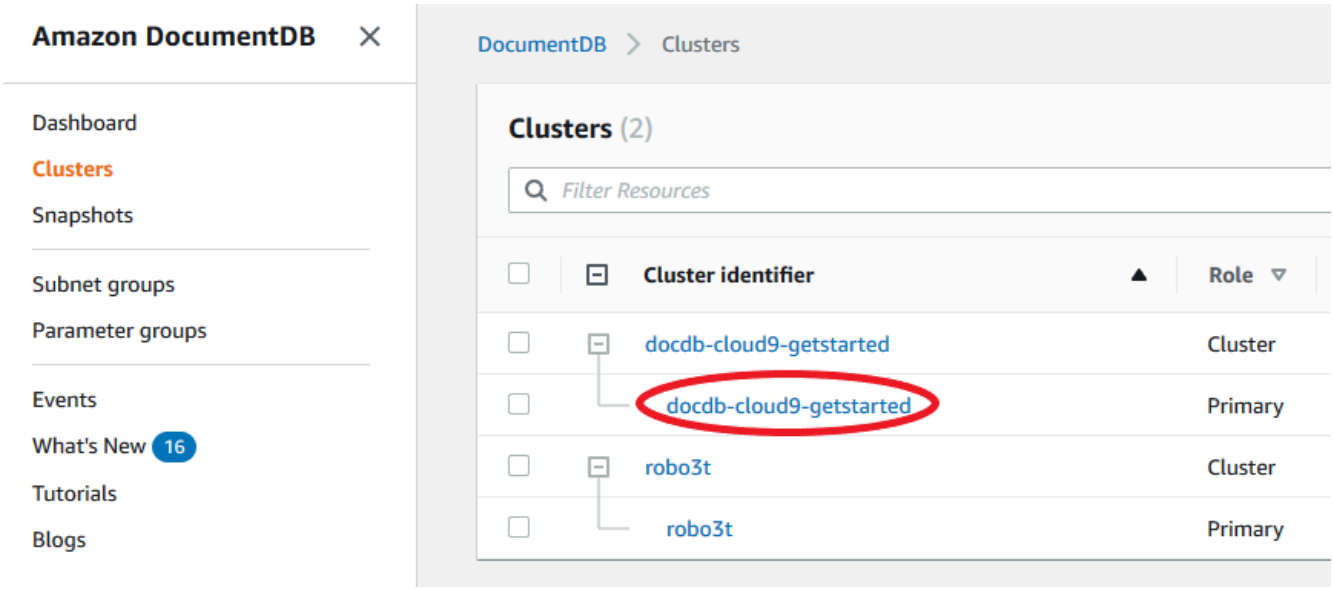
- Amazon DocumentDB 4.0:2.0.10179 oder höher
- Amazon DocumentDB 5.0:3.0.4780 oder höher

Sie können die aktuelle Patch-Version der Amazon DocumentDB DocumentDB-Engine ermitteln, indem Sie den folgenden Befehl ausführen:`db.runCommand({getEngineVersion: 1})`.
Bevor Sie das Serverzertifikat aktualisieren, stellen Sie sicher, dass Sie den Vorgang abgeschlossen Schritt 1 haben.

Using the AWS Management Console

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. Möglicherweise müssen Sie herausfinden, welche Instanzen sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2019`). Sie können dies in der Spalte Zertifizierungsstelle tun, die sich ganz rechts in der Cluster-Tabelle befindet.
5. In der Cluster-Tabelle sehen Sie ganz links die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



6. Markieren Sie das Kästchen links neben der Instanz, an der Sie interessiert sind.
7. Wählen Sie Aktionen und dann Ändern.
8. Wählen Sie unter Certificate authority (Zertifizierungsstelle) das neue Serverzertifikat (d. h. `rds-ca-rsa2048-g1`) für diese Instance aus.
9. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.
10. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden. Wenn Sie beabsichtigen, das Serverzertifikat sofort zu ändern, verwenden Sie die Option Apply Immediately (Sofort anwenden) .
11. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Using the AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS CLI

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Automatische Rotation von Serverzertifikaten

Amazon DocumentDB unterstützt die automatische Rotation von Serverzertifikaten. Das Serverzertifikat ist das Blattzertifikat, das für jede Cluster-Instance ausgestellt wurde. Im Gegensatz zu den Root-CA-Zertifikaten haben die Serverzertifikate eine kurze Gültigkeitsdauer (12 Monate) und Amazon DocumentDB führt ihre Rotation automatisch durch, ohne dass Sie etwas unternehmen müssen. Amazon DocumentDB verwendet dieselbe Root-CA für diese automatische Rotation, sodass Sie kein neues CA-Bundle herunterladen müssen.

> **Important**
>
> Wenn Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen, empfehlen wir, dass Sie dem Root-CA-Bundle vertrauen, anstatt jedem Serverzertifikat direkt zu vertrauen. Dadurch werden Verbindungsfehler vermieden, nachdem das Serverzertifikat rotiert wurde. Siehe Verbindung bei aktiviertem TLS herstellen.

Amazon DocumentDB versucht, Ihr Serverzertifikat in Ihrem bevorzugten Wartungsfenster mit der Halbwertszeit des Serverzertifikats zu rotieren. Das neue Serverzertifikat ist 12 Monate gültig.

Verwenden Sie den `describe-db-engine-versions`Befehl und überprüfen Sie das `SupportsCertificateRotationWithoutRestart` Kennzeichen, um festzustellen, ob die Engine-Version das Rotieren des Zertifikats ohne Neustart unterstützt.

> **Note**
>
> Amazon DocumentDB unterstützt die Rotation von Serverzertifikaten ohne Neustarts, wenn Ihr Cluster auf den folgenden Engine-Patch-Versionen läuft:
>
> - Amazon DocumentDB 3.6:1.0.208662 oder höher
> - Amazon DocumentDB 4.0:2.0.10179 oder höher

- Amazon DocumentDB 5.0:3.0.4780 oder höher

Sie können die aktuelle Patch-Version der Amazon DocumentDB DocumentDB-Engine ermitteln, indem Sie diesen Befehl ausführen:`db.runCommand({getEngineVersion: 1})`.
Wenn Sie eine ältere Engine-Patch-Version verwenden, rotiert Amazon DocumentDB das Serverzertifikat und plant einen Datenbank-Neustart in Ihrem bevorzugten Wartungsfenster.

Fehlerbehebung

Wenn im Rahmen der Zertifikatrotation Probleme beim Herstellen einer Verbindung mit dem Cluster auftreten, empfehlen wir Folgendes:

- Stellen Sie sicher, dass Ihre Clients das neueste Zertifikatpaket verwenden. Siehe Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?.
- Stellen Sie sicher, dass Ihre Instances das neueste Zertifikat verwenden. Siehe Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?.
- Stellen Sie sicher, dass die neueste Zertifizierungsstelle von Ihrer Anwendung verwendet wird. Einige Treiber, wie Java und Go, benötigen zusätzlichen Code, um mehrere Zertifikate aus einem Zertifikatpaket in den Vertrauensspeicher zu importieren. Weitere Informationen zur Verbindung mit Amazon DocumentDB mit TLS finden Sie unterProgrammgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB.
- Wenden Sie sich an den Support. Wenn Sie Fragen oder Probleme haben, wenden Sie sich an Support.

Häufig gestellte Fragen

Im Folgenden finden Sie Antworten auf einige häufig gestellte Fragen zu TLS-Zertifikaten.

Was passiert, wenn ich Fragen oder Probleme habe?

Wenn Sie Fragen oder Probleme haben, wenden Sie sich an Support.

Woher weiß ich, ob ich TLS verwende, um eine Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Sie können bestimmen, ob Ihr Cluster TLS verwendet, indem Sie den `tls`-Parameter für die Clusterparametergruppe Ihres Clusters untersuchen. Wenn der `tls`-Parameter auf `enabled` festgelegt ist, verwenden Sie das TLS-Zertifikat, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

Warum aktualisieren Sie die Zertifizierungsstellen- und Serverzertifikate?

Die Amazon DocumentDB-CA- und Serverzertifikate werden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert.

Was passiert, wenn ich bis zum Ablaufdatum nichts unternehme?

Wenn Sie TLS verwenden, um mit einem abgelaufenen CA-Zertifikat eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, können Ihre Anwendungen, die eine Verbindung über TLS herstellen, nicht mehr mit dem Amazon DocumentDB-Cluster kommunizieren.

Amazon DocumentDB rotiert Ihre Datenbankzertifikate vor Ablauf nicht automatisch. Sie müssen Ihre Anwendungen und Cluster aktualisieren, damit sie die neuen CA-Zertifikate vor oder nach dem Ablaufdatum verwenden können.

Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?

Um die Amazon DocumentDB-Instances zu identifizieren, die noch das alte Serverzertifikat verwenden, können Sie entweder die Amazon DocumentDB AWS Management Console oder die verwenden. AWS CLI

Mit dem AWS Management Console

So identifizieren Sie die Instances in Ihren Clustern, die das ältere Zertifikat verwenden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Instances befinden.

3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. In der Spalte Zertifizierungsstelle (ganz rechts in der Tabelle) wird angezeigt, welche Instanzen sich noch auf dem alten Serverzertifikat (`rds-ca-2019`) und dem neuen Serverzertifikat (`rds-ca-rsa2048-g1`) befinden.

Mit dem AWS CLI

Um die Instances in Ihren Clustern zu identifizieren, die das ältere Serverzertifikat verwenden, verwenden Sie den Befehl `describe-db-clusters` mit Folgendem:

```
aws docdb describe-db-instances \
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Wie ändere ich einzelne Instances in meinem Amazon DocumentDB-Cluster, um das Serverzertifikat zu aktualisieren?

Sie sollten die Serverzertifikate für alle Instances in einem Cluster gleichzeitig aktualisieren. Um die Instances im Cluster zu ändern, können Sie die Konsole oder die AWS CLI verwenden.

> **Note**
>
> Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

Mit dem AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. In der Spalte Zertifizierungsstelle (ganz rechts in der Tabelle) wird angezeigt, welche Instanzen sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2019`).
5. Wählen Sie in der Cluster-Tabelle unter Cluster-ID eine Instanz aus, die geändert werden soll.

6. Wählen Sie Aktionen und dann Ändern.
7. Wählen Sie unter Certificate authority (Zertifizierungsstelle) das neue Serverzertifikat (d. h. `rds-ca-rsa2048-g1`) für diese Instance aus.
8. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.
9. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden.
10. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Verwenden Sie den AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS CLI

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa2048-g1 --no-apply-immediately
```

Was passiert, wenn ich einem vorhandenen Cluster eine neue Instance hinzufüge?

Alle neu erstellten Instances verwenden das alte Serverzertifikat und erfordern TLS-Verbindungen über das alte CA-Zertifikat. Alle neuen Amazon DocumentDB DocumentDB-Instances, die nach dem 25. Januar 2024 erstellt wurden, verwenden standardmäßig das neue Zertifikat rds-ca-rsa 2048-g1.

Was passiert, wenn ein Instance-Ersatz oder ein Failover auf meinem Cluster vorhanden ist?

Wenn ein Instance-Ersatz in Ihrem Cluster vorhanden ist, verwendet die neu erstellte Instance weiterhin dasselbe Serverzertifikat wie die Instance davor. Es wird empfohlen, Serverzertifikate für alle Instances gleichzeitig zu aktualisieren. Wenn ein Failover im Cluster auftritt, wird das Serverzertifikat auf dem neuen Primärserver verwendet.

Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?

Wir empfehlen dringend, TLS zu aktivieren. Für den Fall, dass Sie TLS nicht aktivieren, empfehlen wir dennoch, die Zertifikate auf Ihren Amazon DocumentDB DocumentDB-Instances zu rotieren, falls Sie in future TLS für die Verbindung mit Ihren Clustern verwenden möchten. Wenn Sie nie planen, TLS für die Verbindung zu Ihren Amazon DocumentDB-Clustern zu verwenden, sind keine Maßnahmen erforderlich.

Was soll ich tun, wenn ich derzeit nicht TLS für die Verbindung mit meinem Cluster verwende, dies zukünftig aber vorhabe?

Wenn Sie vor Januar 2024 einen Cluster erstellt haben, folgen Sie Schritt 1 und Schritt 2 im vorherigen Abschnitt, um sicherzustellen, dass Ihre Anwendung das aktualisierte CA-Bundle verwendet und dass jede Amazon DocumentDB DocumentDB-Instance das neueste Serverzertifikat verwendet. Wenn Sie nach dem 25. Januar 2024 einen Cluster erstellen, verfügt Ihr Cluster bereits über das neueste Serverzertifikat (rds-ca-rsa2048-g1). Informationen zum Überprüfen, ob Ihre Anwendung das neueste CA-Paket verwendet, finden Sie unter Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?.

Kann die Frist über den August 2024 hinaus verlängert werden?

Wenn Ihre Anwendungen eine Verbindung über TLS herstellen, kann die Frist nicht verlängert werden.

Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?

Verwenden Sie den folgenden Befehl, um zu überprüfen, ob Sie das neueste Paket haben. Um diesen Befehl ausführen zu können, muss Java installiert sein und die Java-Tools müssen sich in der PATH-Variablen Ihrer Shell befinden. Weitere Informationen finden Sie unter Java verwenden

macOS und Amazon Linux

```
keytool -printcert -v -file global-bundle.pem
```

Windows

```
keytool -printcert -v -file global-bundle.p7b
```

Warum sehe ich „RDS“ im Namen des Zertifizierungsstellenpakets?

Für bestimmte Verwaltungsfunktionen, wie z. B. die Zertifikatsverwaltung, verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.

Wann läuft das neue Zertifikat ab?

Das neue Serverzertifikat läuft (in der Regel) wie folgt ab:

- rds-ca-rsa2048-g1 — Läuft 2061 ab
- rds-ca-rsa4096-g1 — Läuft 2121 ab
- rds-ca-ecc384-g1 — Läuft 2121 ab

Welche Fehler treten auf, wenn ich vor Ablauf des Zertifikats keine Maßnahmen ergreife?

Die Fehlermeldungen variieren je nach Treiber. Im Allgemeinen werden Ihnen Fehler bei der Zertifikatsvalidierung angezeigt, die die Zeichenfolge „Zertifikat ist abgelaufen“ enthalten.

Kann ich nach der Anwendung des neuen Serverzertifikats wieder zum alten Zertifikat zurückkehren?

Wenn Sie eine Instance auf das alte Serverzertifikat zurücksetzen müssen, sollten Sie alle Instances im Cluster zurücksetzen. Sie können das Serverzertifikat für jede Instanz in einem Cluster rückgängig machen, indem Sie den AWS Management Console oder den AWS CLI verwenden.

Mit dem AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. Wählen Sie in der Cluster-Tabelle unter Cluster-ID eine Instanz aus, die Sie ändern möchten. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
5. Unter Certificate authority (Zertifizierungsstelle) können Sie das alte Serverzertifikat (`rds-ca-2019`) auswählen.
6. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.
7. Auf dieser resultierenden Seite können Sie festlegen, dass Ihre Änderungen im nächsten Wartungsfenster oder sofort angewendet werden sollen. Wählen Sie die gewünschte Option und anschließend Modify instance (Instance ändern) aus.

> **Note**
>
> Wenn Sie Ihre Änderungen sofort anwenden möchten, werden alle Änderungen in der Warteschlange für ausstehende Änderungen ebenfalls angewendet. Wenn eine der ausstehenden Änderungen eine Ausfallszeit erfordert, kann die Auswahl dieser Option einen unerwarteten Ausfall verursachen.

Verwenden Sie den AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2019 <--apply-immediately | --no-apply-immediately>
```

Wenn Sie `--no-apply-immediately` wählen, werden die Änderungen während des nächsten Wartungsfensters des Clusters angewendet.

Wird bei einer Wiederherstellung, die ich aus einem Snapshot oder zeitpunktbezogen ausführe, das neue Serverzertifikat verwendet?

Wenn Sie nach August 2024 einen Snapshot point-in-time wiederherstellen oder eine Wiederherstellung durchführen, verwendet der neu erstellte Cluster das neue CA-Zertifikat.

Was ist, wenn ich Probleme habe, von einem beliebigen Mac OS aus eine direkte Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Mac OS hat die Anforderungen für vertrauenswürdige Zertifikate aktualisiert. Vertrauenswürdige Zertifikate müssen jetzt 397 Tage oder weniger gültig sein (siehehttps://support.apple.com/en-us/HT211025).

> **Note**
>
> Diese Einschränkung gilt für neuere Versionen von Mac OS.

Amazon DocumentDB DocumentDB-Instance-Zertifikate sind über vier Jahre gültig und damit länger als das Mac OS-Maximum. Um von einem Computer mit Mac OS aus eine direkte Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, müssen Sie beim Erstellen der TLS-Verbindung ungültige Zertifikate zulassen. In diesem Fall bedeuten ungültige Zertifikate, dass die Gültigkeitsdauer mehr als 397 Tage beträgt. Sie sollten die Risiken verstehen, bevor Sie ungültige Zertifikate zulassen, wenn Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Verwenden Sie den Parameter, um von Mac OS aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen AWS CLI, indem Sie den `tlsAllowInvalidCertificates` Parameter verwenden.

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate — GovCloud

Themen

- Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters
- Fehlerbehebung
- Häufig gestellte Fragen

> **Note**
>
> Diese Informationen gelten für Benutzer in den Regionen GovCloud (US-West) und GovCloud (US-Ost).

Das Zertifikat der Zertifizierungsstelle (CA) für Amazon DocumentDB-Cluster (mit MongoDB-Kompatibilität) wird am 18. Mai 2022 aktualisiert. Wenn Sie Amazon DocumentDB-Cluster mit aktiviertem Transport Layer Security (TLS) verwenden (Standardeinstellung) und Sie Ihre Client-Anwendungs- und Serverzertifikate nicht rotiert haben, sind die folgenden Schritte erforderlich, um Verbindungsprobleme zwischen Ihrer Anwendung und Ihren Amazon DocumentDB-Clustern zu beheben.

- Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung
- Schritt 2: Aktualisieren des Serverzertifikats

Die CA- und Serverzertifikate wurden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Das vorherige CA-Zertifikat läuft am 18. Mai 2022 ab. Client-Anwendungen müssen die neuen CA-Zertifikate zu ihren Trust Stores hinzufügen, und bestehende Amazon DocumentDB DocumentDB-Instances müssen aktualisiert werden, sodass sie die neuen CA-Zertifikate vor diesem Ablaufdatum verwenden können.

Aktualisierung Ihrer Anwendung und Ihres Amazon DocumentDB-Clusters

Führen Sie die Schritte in diesem Abschnitt aus, um das CA-Zertifikatspaket Ihrer Anwendung (Schritt 1) und die Serverzertifikate Ihres Clusters (Schritt 2) zu aktualisieren. Bevor Sie die Änderungen auf Ihre Produktionsumgebungen anwenden, empfehlen wir dringend, diese Schritte in einer Entwicklungs- oder Stagingumgebung zu testen.

> **Note**
>
> Sie müssen die Schritte 1 und 2 jeweils AWS-Region ausführen, wenn Sie Amazon DocumentDB-Cluster haben.

Schritt 1: Herunterladen des neuen CA-Zertifikats und Aktualisieren Ihrer Anwendung

Laden Sie das neue CA-Zertifikat herunter und aktualisieren Sie Ihre Anwendung so, dass sie das neue CA-Zertifikat verwendet, um TLS-Verbindungen zu Amazon DocumentDB in Ihrer spezifischen Region herzustellen:

- Laden Sie für GovCloud (US-West) das neue CA-Zertifikatspaket von herunter. https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem Diese Operation lädt die Datei mit dem Namen `us-gov-west-1-bundle.pem` herunter.
- Laden Sie für GovCloud (US-Ost) das neue CA-Zertifikatspaket von herunter. https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-east-1/us-gov-east-1-bundle.pem Diese Operation lädt die Datei mit dem Namen `us-gov-east-1-bundle.pem` herunter.

> **Note**
>
> Wenn Sie auf den Keystore zugreifen, der sowohl das alte CA-Zertifikat (`rds-ca-2017-root.pem`) als auch die neuen CA-Zertifikate (`rds-ca-rsa2048-g1.pem`,, oder`rds-ca-ecc384-g1.pem`) enthält, stellen Sie sicher`rds-ca-rsa4096-g1.pem`, dass der Keystore das Zertifikat Ihrer Wahl auswählt. Einzelheiten zu den einzelnen Zertifikaten finden Sie in Schritt 2 unten.

```
wget https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-west-1/us-gov-west-1-bundle.pem
```

```
wget https://truststore.pki.us-gov-west-1.rds.amazonaws.com/us-gov-east-1/us-gov-east-1-bundle.pem
```

Aktualisieren Sie als Nächstes Ihre Anwendungen, um das neue Zertifikatpaket zu verwenden. Das neue CA-Bundle enthält sowohl das alte CA-Zertifikat als auch das neue CA-Zertifikat (`rds-ca-rsa2048-g1.pem`,`rds-ca-rsa4096-g1.pem`, oder`rds-ca-ecc384-g1.pem`). Wenn beide CA-Zertifikate im neuen CA-Paket enthalten sind, können Sie Ihre Anwendung und Ihren Cluster in zwei Schritten aktualisieren.

Für alle Downloads des CA-Zertifikatspakets nach dem 21. Dezember 2021 sollte das neue CA-Zertifikatspaket verwendet werden. Informationen dazu, wie Sie überprüfen, ob Ihre Anwendung das neueste CA-Zertifikatspaket verwendet, finden Sie unter Wie kann ich sicher sein, dass ich das

neueste Zertifizierungsstellenpaket verwende?. Wenn Sie das neueste CA-Zertifikatspaket in Ihrer Anwendung bereits verwenden, können Sie mit Schritt 2 fortfahren.

Beispiele für die Verwendung eines Zertifizierungsstellenpakets mit Ihrer Anwendung finden Sie unter Verschlüsseln von Daten während der Übertragung und Verbindung bei aktiviertem TLS herstellen.

> **Note**
>
> Zurzeit akzeptiert der MongoDB Go Driver 1.2.1 nur ein einzelnes CA-Serverzertifikat in `sslcertificateauthorityfile`. Informationen dazu, wie Sie bei aktiviertem TLS eine Verbindung zu Amazon DocumentDB über Go herstellen, finden Sie unter Verbindung bei aktiviertem TLS herstellen.

Schritt 2: Aktualisieren des Serverzertifikats

Nachdem die Anwendung für die Verwendung des neuen CA-Bundles aktualisiert wurde, besteht der nächste Schritt darin, das Serverzertifikat zu aktualisieren, indem jede Instance in einem Amazon DocumentDB-Cluster geändert wird. Informationen zum Ändern von Instances zur Verwendung des neuen Serverzertifikats finden Sie in den folgenden Anweisungen.

Amazon DocumentDB bietet Folgendes CAs , um das DB-Serverzertifikat für eine DB-Instance zu signieren:

- rds-ca-ecc384-g1 — Verwendet eine Zertifizierungsstelle mit ECC 384-Algorithmus für private Schlüssel und Signaturalgorithmus. SHA384 Diese CA unterstützt die automatische Rotation von Serverzertifikaten. Dies wird nur auf Amazon DocumentDB 4.0 und 5.0 unterstützt.
- rds-ca-rsa2048-g1 — Verwendet in den meisten Regionen eine Zertifizierungsstelle mit dem RSA 2048-Algorithmus für private Schlüssel und dem Signaturalgorithmus. SHA256 AWS Diese CA unterstützt die automatische Rotation von Serverzertifikaten.
- rds-ca-rsa4096-g1 — Verwendet eine Zertifizierungsstelle mit dem RSA 4096-Algorithmus für private Schlüssel und dem Signaturalgorithmus. SHA384 Diese CA unterstützt die automatische Rotation von Serverzertifikaten.

> **Note**
>
> Wenn Sie den verwenden AWS CLI, können Sie die Gültigkeiten der oben aufgeführten Zertifizierungsstellen mithilfe von Describe-Zertifikaten überprüfen.

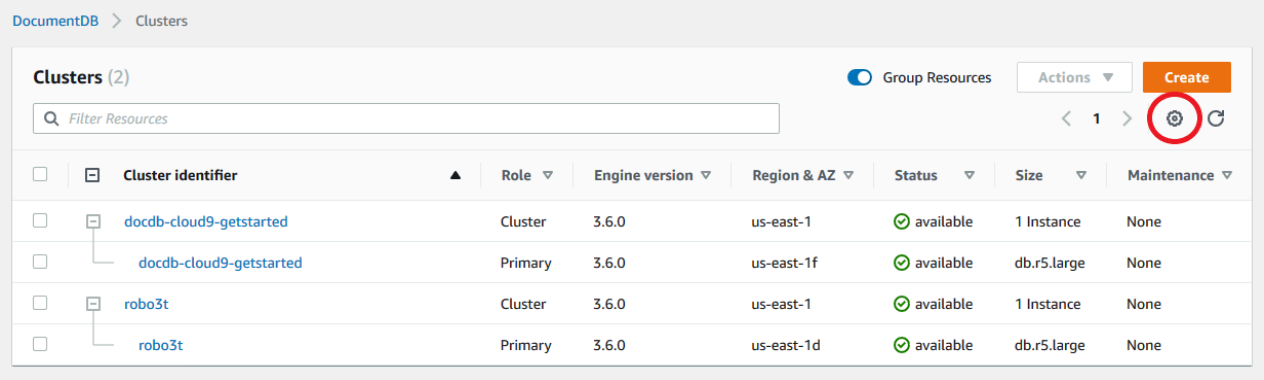
> **Note**
>
> Amazon DocumentDB 4.0- und 5.0-Instances erfordern keinen Neustart.
> Für die Aktualisierung Ihrer Amazon DocumentDB 3.6-Instances ist ein Neustart erforderlich, was zu Serviceunterbrechungen führen kann. Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

Using the AWS Management Console

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Clusters aus.
4. Möglicherweise müssen Sie herausfinden, welche Instanzen sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2017`). Sie können dies in der Spalte Zertifizierungsstelle tun, die standardmäßig ausgeblendet ist. Gehen Sie folgendermaßen vor, um die Spalte Certificate authority (Zertifizierungsstelle) anzuzeigen:

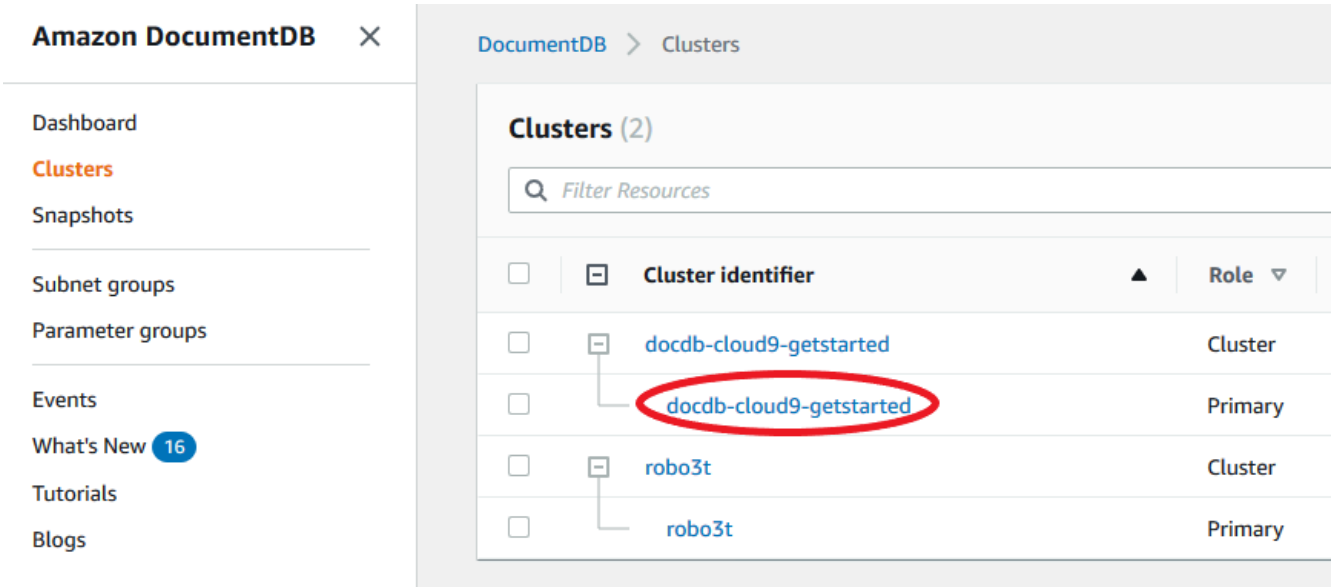
   a. Wählen Sie das Symbol Settings (Einstellungen).



b. Wählen Sie in der Liste der sichtbaren Spalten die Spalte Certificate authority (Zertifizierungsstelle).

c. Wählen Sie Confirm (Bestätigen), um Ihre Änderungen zu speichern.

5. Jetzt sehen Sie wieder im Cluster-Navigationsfeld die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



6. Markieren Sie das Kästchen links neben der Instanz, an der Sie interessiert sind.

7. Wählen Sie Aktionen und dann Ändern.

8. Wählen Sie unter Zertifizierungsstelle das neue Serverzertifikat (`rds-ca-rsa2048-g1rds-ca-rsa4096-g1`, oder`rds-ca-ecc384-g1`) für diese Instanz aus.

9. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.

10. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden. Wenn Sie beabsichtigen, das Serverzertifikat sofort zu ändern, verwenden Sie die Option Apply Immediately (Sofort anwenden) .
11. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Using the AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS CLI

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus. Verwenden Sie eines der folgenden Zertifikate:`rds-ca-rsa2048-g1`,`rds-ca-rsa4096-g1`, oder`rds-ca-ecc384-g1`.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus. Verwenden Sie eines der folgenden Zertifikate:`rds-ca-rsa2048-g1`,`rds-ca-rsa4096-g1`, oder`rds-ca-ecc384-g1`.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier>
 --ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Fehlerbehebung

Wenn im Rahmen der Zertifikatrotation Probleme beim Herstellen einer Verbindung mit dem Cluster auftreten, empfehlen wir Folgendes:

- Starten Sie Ihre Instances neu. Das Rotieren des neuen Zertifikats erfordert, dass Sie jede Ihrer Instances neu starten. Wenn Sie das neue Zertifikat auf eine oder mehrere Instances angewendet haben, diese jedoch nicht neu gestartet haben, starten Sie die Instances neu, um das neue Zertifikat anzuwenden. Weitere Informationen finden Sie unter Eine Amazon DocumentDB DocumentDB-Instance neu starten.
- Stellen Sie sicher, dass Ihre Clients das neueste Zertifikatpaket verwenden. Siehe Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?.

- Stellen Sie sicher, dass Ihre Instances das neueste Zertifikat verwenden. Siehe Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?.
- Stellen Sie sicher, dass die neueste Zertifizierungsstelle von Ihrer Anwendung verwendet wird. Einige Treiber, wie Java und Go, benötigen zusätzlichen Code, um mehrere Zertifikate aus einem Zertifikatpaket in den Vertrauensspeicher zu importieren. Weitere Informationen zur Verbindung mit Amazon DocumentDB mit TLS finden Sie unterProgrammgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB.
- Wenden Sie sich an den Support. Wenn Sie Fragen oder Probleme haben, wenden Sie sich an Support.

Häufig gestellte Fragen

Im Folgenden finden Sie Antworten auf einige häufig gestellte Fragen zu TLS-Zertifikaten.

Was passiert, wenn ich Fragen oder Probleme habe?

Wenn Sie Fragen oder Probleme haben, wenden Sie sich an Support.

Woher weiß ich, ob ich TLS verwende, um eine Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Sie können bestimmen, ob Ihr Cluster TLS verwendet, indem Sie den `tls`-Parameter für die Clusterparametergruppe Ihres Clusters untersuchen. Wenn der `tls`-Parameter auf `enabled` festgelegt ist, verwenden Sie das TLS-Zertifikat, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

Warum aktualisieren Sie die Zertifizierungsstellen- und Serverzertifikate?

Die Amazon DocumentDB-CA- und Serverzertifikate wurden im Rahmen der bewährten Standardmethoden für Wartung und Sicherheit für Amazon DocumentDB aktualisiert. Die aktuellen CA- und Serverzertifikate laufen am Mittwoch, den 18. Mai 2022 ab.

Was passiert, wenn ich bis zum Ablaufdatum keine Maßnahmen ergreife?

Wenn Sie TLS verwenden, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, und Sie die Änderung nicht bis zum 18. Mai 2022 vornehmen, können Ihre Anwendungen, die eine Verbindung über TLS herstellen, nicht mehr mit dem Amazon DocumentDB-Cluster kommunizieren.

Amazon DocumentDB rotiert Ihre Datenbankzertifikate vor Ablauf nicht automatisch. Sie müssen Ihre Anwendungen und Cluster aktualisieren, damit sie die neuen CA-Zertifikate vor oder nach dem Ablaufdatum verwenden können.

Woher weiß ich, welche meiner Amazon DocumentDB DocumentDB-Instances das alte/neue Serverzertifikat verwenden?

Um die Amazon DocumentDB-Instances zu identifizieren, die noch das alte Serverzertifikat verwenden, können Sie entweder die Amazon DocumentDB AWS Management Console oder die verwenden. AWS CLI

Unter Verwendung der AWS Management Console

So identifizieren Sie die Instances in Ihren Clustern, die das ältere Zertifikat verwenden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Instances befinden.
3. Klicken Sie im Navigationsbereich links in der Konsole auf Instances (Instances).
4. In der Spalte Zertifizierungsstelle (standardmäßig ausgeblendet) wird angezeigt, welche Instanzen sich noch auf dem alten Serverzertifikat (`rds-ca-2017`) und dem neuen Serverzertifikat (`rds-ca-rsa2048-g1rds-ca-rsa4096-g1`, oder) befinden. `rds-ca-ecc384-g1` Gehen Sie folgendermaßen vor, um die Spalte Certificate authority (Zertifizierungsstelle) anzuzeigen:
   a. Wählen Sie das Symbol Settings (Einstellungen).
   b. Wählen Sie in der Liste der sichtbaren Spalten die Spalte Certificate authority (Zertifizierungsstelle).
   c. Wählen Sie Confirm (Bestätigen), um Ihre Änderungen zu speichern.

Unter Verwendung des AWS CLI

Um die Instances in Ihren Clustern zu identifizieren, die das ältere Serverzertifikat verwenden, verwenden Sie den Befehl `describe-db-clusters` mit Folgendem:

```
aws docdb describe-db-instances \
```

```
    --filters Name=engine,Values=docdb \
    --query 'DBInstances[*].
{CertificateVersion:CACertificateIdentifier,InstanceID:DBInstanceIdentifier}'
```

Wie ändere ich einzelne Instances in meinem Amazon DocumentDB-Cluster, um das Serverzertifikat zu aktualisieren?

Sie sollten die Serverzertifikate für alle Instances in einem Cluster gleichzeitig aktualisieren. Um die Instances im Cluster zu ändern, können Sie die Konsole oder die AWS CLI verwenden.

> **Note**
>
> Das Aktualisieren Ihrer Instances erfordert einen Neustart, der zu Unterbrechungen des Service führen kann. Sie müssen Schritt 1 abschließen, bevor Sie das Serverzertifikat aktualisieren.

Mit dem AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Klicken Sie im Navigationsbereich links in der Konsole auf Instances (Instances).
4. Die Spalte Certificate authority (Zertifizierungsstelle) (standardmäßig ausgeblendet) zeigt an, welche Instances sich noch auf dem alten Serverzertifikat befinden (`rds-ca-2017`). Gehen Sie folgendermaßen vor, um die Spalte Certificate authority (Zertifizierungsstelle) anzuzeigen:
   a. Wählen Sie das Symbol Settings (Einstellungen).
   b. Wählen Sie in der Liste der sichtbaren Spalten die Spalte Certificate authority (Zertifizierungsstelle).
   c. Wählen Sie Confirm (Bestätigen), um Ihre Änderungen zu speichern.
5. Wählen Sie eine zu ändernde Instance aus.
6. Wählen Sie Aktionen und dann Ändern.
7. Wählen Sie unter Zertifizierungsstelle eines der neuen Serverzertifikate (`rds-ca-rsa2048-g1rds-ca-rsa4096-g1`, oder`rds-ca-ecc384-g1`) für diese Instanz aus.

8. Sie können eine Zusammenfassung der Änderungen auf der nächsten Seite sehen. Beachten Sie, dass es eine zusätzliche Warnung gibt, die Sie daran erinnert, dass Ihre Anwendung das neueste Zertifizierungsstellenpaket verwendet, bevor Sie die Instance ändern, um eine Unterbrechung der Konnektivität zu vermeiden.
9. Sie können die Änderung während Ihres nächsten Wartungsfensters oder sofort anwenden.
10. Wählen Sie Modify instance (Instance ändern), um die Aktualisierung abzuschließen.

Unter Verwendung des AWS CLI

Führen Sie die folgenden Schritte aus, um das alte Serverzertifikat für Ihre vorhandenen Amazon DocumentDB DocumentDB-Instances mithilfe von zu identifizieren und zu rotieren. AWS CLI

1. Um die Instances sofort zu ändern, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --apply-immediately
```

2. Um die Instances in Ihren Clustern so zu ändern, dass sie während des nächsten Wartungsfensters Ihres Clusters das neue CA-Zertifikat verwenden, führen Sie für jede Instance im Cluster den folgenden Befehl aus.

```
aws docdb modify-db-instance --db-instance-identifier <yourInstanceIdentifier> --
ca-certificate-identifier rds-ca-rsa4096-g1 --no-apply-immediately
```

Was passiert, wenn ich einem vorhandenen Cluster eine neue Instance hinzufüge?

Alle neu erstellten Instances verwenden das alte Serverzertifikat und erfordern TLS-Verbindungen über das alte CA-Zertifikat. Alle neuen Amazon DocumentDB DocumentDB-Instances, die nach dem 21. März 2022 erstellt wurden, verwenden standardmäßig die neuen Zertifikate.

Was passiert, wenn ein Instance-Ersatz oder ein Failover auf meinem Cluster vorhanden ist?

Wenn ein Instance-Ersatz in Ihrem Cluster vorhanden ist, verwendet die neu erstellte Instance weiterhin dasselbe Serverzertifikat wie die Instance davor. Es wird empfohlen, Serverzertifikate

für alle Instances gleichzeitig zu aktualisieren. Wenn ein Failover im Cluster auftritt, wird das Serverzertifikat auf dem neuen Primärserver verwendet.

Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?

Wenn Sie TLS nicht verwenden, um eine Verbindung zu Ihren Amazon DocumentDB-Clustern herzustellen, sind keine Maßnahmen erforderlich.

Was soll ich tun, wenn ich derzeit nicht TLS für die Verbindung mit meinem Cluster verwende, dies zukünftig aber vorhabe?

Wenn Sie vor dem 21. März 2022 einen Cluster erstellt haben, folgen Sie Schritt 1 und Schritt 2 im vorherigen Abschnitt, um sicherzustellen, dass Ihre Anwendung das aktualisierte CA-Bundle verwendet und dass jede Amazon DocumentDB DocumentDB-Instance das neueste Serverzertifikat verwendet. Wenn Sie nach dem 21. März 2022 einen Cluster erstellen, verfügt Ihr Cluster bereits über das neueste Serverzertifikat. Informationen zum Überprüfen, ob Ihre Anwendung das neueste CA-Paket verwendet, finden Sie unter Wenn ich keine TLS für die Verbindung mit meinem Cluster verwende, muss ich trotzdem jede meiner Instances aktualisieren?.

Kann die Frist über den 18. Mai 2022 hinaus verlängert werden?

Wenn Ihre Anwendungen eine Verbindung über TLS herstellen, kann die Frist nicht über den 18. Mai 2022 hinaus verlängert werden.

Wie kann ich sicher sein, dass ich das neueste Zertifizierungsstellenpaket verwende?

Aus Kompatibilitätsgründen werden sowohl alte als auch neue Zertifizierungsstellenpaket-Dateien mit `us-gov-west-1-bundle.pem` benannt. Darüber hinaus können Sie Tools wie `openssl` oder `keytool` verwenden, um das CA-Paket zu überprüfen.

Warum sehe ich „RDS“ im Namen des Zertifizierungsstellenpakets?

Für bestimmte Verwaltungsfunktionen, wie z. B. die Zertifikatsverwaltung, verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.

Wann läuft das neue Zertifikat ab?

Das neue Serverzertifikat läuft (in der Regel) wie folgt ab:

- rds-ca-rsa2048-g1 — Läuft 2061 ab
- rds-ca-rsa4096-g1 — Läuft 2121 ab
- rds-ca-ecc384-g1 — Läuft 2121 ab

Welche Fehler treten auf, wenn ich vor Ablauf des Zertifikats keine Maßnahmen ergreife?

Die Fehlermeldungen variieren je nach Treiber. Im Allgemeinen werden Ihnen Fehler bei der Zertifikatsvalidierung angezeigt, die die Zeichenfolge „Zertifikat ist abgelaufen“ enthalten.

Kann ich nach der Anwendung des neuen Serverzertifikats wieder zum alten Zertifikat zurückkehren?

Wenn Sie eine Instance auf das alte Serverzertifikat zurücksetzen müssen, sollten Sie alle Instances im Cluster zurücksetzen. Sie können das Serverzertifikat für jede Instanz in einem Cluster rückgängig machen, indem Sie den AWS Management Console oder den AWS CLI verwenden.

Mit dem AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Liste der Regionen in der oberen rechten Ecke des Bildschirms die Region aus, AWS-Region in der sich Ihre Cluster befinden.
3. Klicken Sie im Navigationsbereich links in der Konsole auf Instances (Instances).
4. Wählen Sie eine zu ändernde Instance aus. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
5. Unter Zertifizierungsstelle können Sie das alte Serverzertifikat () auswählen. `rds-ca-2017`
6. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.
7. Auf dieser resultierenden Seite können Sie festlegen, dass Ihre Änderungen im nächsten Wartungsfenster oder sofort angewendet werden sollen. Wählen Sie die gewünschte Option und anschließend Modify instance (Instance ändern) aus.

> **Note**
>
> Wenn Sie Ihre Änderungen sofort anwenden möchten, werden alle Änderungen in der Warteschlange für ausstehende Änderungen ebenfalls angewendet. Wenn eine der

ausstehenden Änderungen eine Ausfallszeit erfordert, kann die Auswahl dieser Option einen unerwarteten Ausfall verursachen.

Unter Verwendung des AWS CLI

```
aws docdb modify-db-instance --db-instance-identifier <db_instance_name> ca-
certificate-identifier rds-ca-2017 <--apply-immediately | --no-apply-immediately>
```

Wenn Sie `--no-apply-immediately` wählen, werden die Änderungen während des nächsten Wartungsfensters des Clusters angewendet.

Wird bei einer Wiederherstellung, die ich aus einem Snapshot oder zeitpunktbezogen ausführe, das neue Serverzertifikat verwendet?

Wenn Sie nach dem 21. März 2022 einen Snapshot point-in-time wiederherstellen oder eine Wiederherstellung durchführen, verwendet der neue Cluster, der erstellt wird, das neue CA-Zertifikat.

Was ist, wenn ich Probleme habe, von Mac OS X Catalina aus eine direkte Verbindung zu meinem Amazon DocumentDB-Cluster herzustellen?

Mac OS X Catalina hat die Anforderungen für vertrauenswürdige Zertifikate aktualisiert. Vertrauenswürdige Zertifikate müssen jetzt 825 Tage oder weniger gültig sein (siehe). https://support.apple.com/en-us/HT210176 Amazon DocumentDB DocumentDB-Instance-Zertifikate sind über vier Jahre gültig und damit länger als das Mac OS X-Maximum. Um von einem Computer mit Mac OS X Catalina aus eine direkte Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, müssen Sie beim Erstellen der TLS-Verbindung ungültige Zertifikate zulassen. In diesem Fall bedeuten ungültige Zertifikate, dass die Gültigkeitsdauer länger als 825 Tage beträgt. Sie sollten die Risiken verstehen, bevor Sie ungültige Zertifikate zulassen, wenn Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Verwenden Sie den Parameter, um über OS X Catalina eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen. AWS CLI`tlsAllowInvalidCertificates`

```
mongo --tls --host <hostname> --username <username> --password <password> --port 27017
 --tlsAllowInvalidCertificates
```

Konformitätsprüfung in Amazon DocumentDB

Die Sicherheit und Konformität von Amazon DocumentDB wird von externen Prüfern im Rahmen mehrerer AWS Compliance-Programme bewertet, darunter die folgenden:

- System and Organization Controls (SOC) 1, 2 und 3. Weitere Informationen finden Sie unter SOC.
- Föderales Risiko- und Autorisierungsmanagementprogramm (FedRAMP). Weitere Informationen finden Sie unterAWS -Services im Rahmen des Compliance-Programms.
- Payment Card Industry Data Security Standard (PCI DSS). Weitere Informationen finden Sie unter PCI DSS.
- ISO 9001, 27001, 27017 und 27018. Weitere Informationen finden Sie unter ISO-zertifiziert.
- Health Insurance Portability and Accountability Act Business Associate Agreement (HIPAA BAA). Weitere Informationen finden Sie unter HIPAA-Compliance.

AWS bietet unter AWS Services in Scope by Compliance Program eine häufig aktualisierte Liste der AWS Services im Rahmen bestimmter Compliance-Programme.

Prüfberichte von Drittanbietern stehen Ihnen zum Herunterladen zur Verfügung unter AWS Artifact. Weitere Informationen finden Sie unter Berichte in AWS Artifact herunterladen.

Weitere Informationen zu AWS Compliance-Programmen finden Sie unter AWS Compliance-Programme.

Ihre Compliance-Verantwortung bei der Verwendung von Amazon DocumentDB richtet sich nach der Sensibilität Ihrer Daten, den Compliance-Zielen Ihrer Organisation und den geltenden Gesetzen und Vorschriften. Wenn Ihre Nutzung von Amazon DocumentDB der Einhaltung von Standards wie HIPAA oder PCI unterliegt, AWS bietet Ihnen folgende Ressourcen:

- AWS Ressourcen zur Einhaltung von Vorschriften — Eine Sammlung von Arbeitsmappen und Leitfäden, die möglicherweise auf Ihre Branche und Ihren Standort zutreffen.
- Schnellstartanleitungen zu Sicherheit und Compliance — Implementierungsleitfäden, in denen architektonische Überlegungen erörtert und Schritte für die Implementierung von Umgebungen beschrieben werden, auf denen Sicherheit und Compliance im Vordergrund stehen. AWS
- AWS Config — Ein Service, der bewertet, wie gut Ihre Ressourcenkonfigurationen internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- AWS Security Hub — Ein umfassender Überblick über Ihren Sicherheitsstatus innerhalb AWS , anhand dessen Sie überprüfen können, ob Sie die Sicherheitsstandards und Best Practices der Sicherheitsbranche einhalten.
- Whitepaper „Architecting for HIPAA Security and Compliance“ — Ein Whitepaper, in dem beschrieben wird, wie Unternehmen HIPAA-konforme Anwendungen entwickeln können AWS .

Resilienz in Amazon DocumentDB

Die AWS globale Infrastruktur basiert auf AWS-Regionen Availability Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Ein Amazon DocumentDB-Cluster kann nur in einer Amazon VPC erstellt werden, die über mindestens zwei Subnetze in mindestens zwei Availability Zones verfügt. Durch die Verteilung Ihrer Cluster-Instances auf mindestens zwei Availability Zones trägt Amazon DocumentDB dazu bei, dass für den unwahrscheinlichen Fall eines Ausfalls der Availability Zone Instances in Ihrem Cluster verfügbar sind. Das Cluster-Volume für Ihren Amazon DocumentDB-Cluster erstreckt sich immer über drei Availability Zones, um dauerhaften Speicher mit geringerer Wahrscheinlichkeit von Datenverlusten bereitzustellen.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur bietet Amazon DocumentDB mehrere Funktionen, um Ihre Anforderungen an Datenstabilität und Datensicherung zu erfüllen.

Fehlertolerante Speicherung, die Probleme automatisch behebt

> Jeder 10-GB-Teil Ihres Speichervolumens wird auf sechs Arten in drei Availability Zones repliziert. Amazon DocumentDB verwendet fehlertoleranten Speicher, der den Verlust von bis zu zwei Datenkopien transparent handhabt, ohne die Schreibverfügbarkeit der Datenbank zu beeinträchtigen, und von bis zu drei Kopien, ohne die Leseverfügbarkeit zu beeinträchtigen. Der Amazon DocumentDB DocumentDB-Speicher repariert sich ebenfalls selbst. Datenblöcke und Festplatten werden kontinuierlich auf Fehler überprüft und automatisch ersetzt.

Manuelle Sicherungen und Wiederherstellung

Amazon DocumentDB bietet die Möglichkeit, vollständige Backups Ihres Clusters für die langfristige Aufbewahrung und Wiederherstellung zu erstellen. Weitere Informationen finden Sie unter Sichern und Wiederherstellen in Amazon DocumentDB.

Point-in-time Wiederherstellung

Point-in-time Recovery schützt Ihre Amazon DocumentDB-Cluster vor versehentlichen Schreib- oder Löschvorgängen. Mit point-in-time Recovery müssen Sie sich keine Gedanken über die Erstellung, Verwaltung oder Planung von On-Demand-Backups machen. Weitere Informationen finden Sie unter Zu einem bestimmten Zeitpunkt wiederherstellen.

Infrastruktursicherheit in Amazon DocumentDB

Als verwalteter Service ist Amazon DocumentDB durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon DocumentDB 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 Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Sie können diese API-Vorgänge von einem beliebigen Netzwerkstandort aus aufrufen. Sie können Amazon DocumentDB DocumentDB-Richtlinien verwenden, um den Zugriff von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf eine bestimmte Amazon DocumentDB DocumentDB-Ressource effektiv nur von der spezifischen VPC innerhalb des Netzwerks isoliert. AWS

> **Note**
>
> Amazon DocumentDB unterstützt keine ressourcenbasierten Zugriffsrichtlinien.

Amazon DocumentDB DocumentDB-API und VPC-Schnittstellen-Endpunkte ()AWS PrivateLink

> **Note**
>
> Elastische Amazon DocumentDB DocumentDB-Cluster unterstützen keine AWS PrivateLink VPC-Endpunkte.

Sie können eine private Verbindung zwischen Ihrer VPC und den Amazon DocumentDB DocumentDB-API-Endpunkten herstellen, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen. Schnittstellen-Endpunkte werden betrieben von. AWS PrivateLink

Instanzbasierte Cluster von Amazon DocumentDB benötigen zwar keine VPC-Endpunktverbindung mit einer Schnittstelle, AWS PrivateLink ermöglichen Ihnen aber den privaten Zugriff auf Amazon DocumentDB DocumentDB-API-Operationen ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder Verbindung. AWS Direct Connect Amazon DocumentDB DocumentDB-Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit Amazon DocumentDB DocumentDB-API-Endpunkten zu kommunizieren und Datenbank-Instances und Datenbankcluster zu starten, zu ändern oder zu beenden. Ihre Amazon DocumentDB DocumentDB-Instances benötigen auch keine öffentlichen IP-Adressen, um die verfügbaren Amazon DocumentDB DocumentDB-API-Operationen nutzen zu können. Der Datenverkehr zwischen Ihrer VPC und Amazon DocumentDB verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic Network-Schnittstellen in Ihren Subnetzen dargestellt. Weitere Informationen finden Sie unter Elastic Network Interfaces im EC2 Amazon-Benutzerhandbuch.

Weitere Informationen zu VPC-Endpunkten finden Sie unter Zugreifen und AWS-Service Verwenden eines VPC-Endpunkts mit einer Schnittstelle im Amazon Virtual Private Cloud (AWS PrivateLink) - Benutzerhandbuch. Weitere Informationen zu Amazon DocumentDB DocumentDB-Vorgängen finden Sie unterReferenz zur Amazon DocumentDB-Cluster-, Instance- und Ressourcenmanagement-API.

Themen

- Überlegungen zu VPC-Endpunkten
- Verfügbarkeit in Regionen
- Erstellen eines VPC-Schnittstellen-Endpunkts für die Amazon DocumentDB DocumentDB-API
- Erstellen einer VPC-Endpunktrichtlinie für die Amazon DocumentDB DocumentDB-API

Überlegungen zu VPC-Endpunkten

Bevor Sie einen VPC-Schnittstellen-Endpunkt für Amazon DocumentDB DocumentDB-API-Endpunkte einrichten, stellen Sie sicher, dass Sie die Voraussetzungen für Schnittstellen-Endpunkte im Amazon Virtual Private Cloud (AWS PrivateLink) -Benutzerhandbuch lesen.

Alle Amazon DocumentDB DocumentDB-API-Operationen, die für die Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen relevant sind, sind über Ihre VPC verfügbar unter. AWS PrivateLink

VPC-Endpunktrichtlinien werden für Amazon DocumentDB DocumentDB-API-Endpunkte unterstützt. Standardmäßig ist der vollständige Zugriff auf Amazon DocumentDB DocumentDB-API-Operationen über den Endpunkt erlaubt. Weitere Informationen finden Sie unter Steuern des Zugriffs auf VPC-Endpunkte mithilfe von Endpunktrichtlinien im Amazon Virtual Private Cloud (AWS PrivateLink) -Benutzerhandbuch.

Verfügbarkeit in Regionen

Die Amazon DocumentDB DocumentDB-API unterstützt derzeit VPC-Endpunkte in den folgenden Bereichen: AWS-Regionen

- US East (Ohio)
- USA Ost (Nord-Virginia)
- USA West (Oregon)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Osaka)

- Asien-Pazifik (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Canada (Central)
- China (Peking)
- China (Ningxia)
- Europe (Frankfurt)
- Europa (Irland)
- Europa (London)
- Europa (Paris)
- Europa (Spain)
- Europa (Milan)
- Naher Osten (VAE)
- Südamerika (São Paulo)
- AWS GovCloud (USA-Ost)
- AWS GovCloud (US-West)

Erstellen eines VPC-Schnittstellen-Endpunkts für die Amazon DocumentDB DocumentDB-API

Sie können einen VPC-Endpunkt für die Amazon DocumentDB DocumentDB-API entweder mit der Amazon VPC-Konsole oder mit () erstellen. AWS Command Line Interface AWS CLI Weitere Informationen finden Sie unter Zugreifen und AWS-Service Verwenden eines VPC-Endpunkts mit einer Schnittstelle im Amazon Virtual Private Cloud (AWS PrivateLink) -Benutzerhandbuch.

Erstellen Sie mithilfe des Servicenamens einen VPC-Endpunkt für die Amazon DocumentDB DocumentDB-API. `com.amazonaws.region.rds`

Mit Ausnahme AWS-Regionen von China können Sie, wenn Sie privates DNS für den Endpunkt aktivieren, API-Anfragen an Amazon DocumentDB mit dem VPC-Endpunkt stellen, indem Sie dessen Standard-DNS-Namen für den Endpunkt verwenden AWS-Regionen, zum Beispiel rds.us-east-1.amazonaws.com. Für China (Peking) und China (Ningxia) AWS-Regionen können Sie API-

Anfragen mit dem VPC-Endpunkt über rds-api.cn-north-1.amazonaws.com .cn bzw. rds-api.cn-northwest-1.amazonaws.com .cn stellen.

Weitere Informationen finden Sie unter Zugreifen und AWS-Service Verwenden eines VPC-Endpunkts mit einer Schnittstelle im Amazon Virtual Private Cloud (AWS PrivateLink) - Benutzerhandbuch.

Erstellen einer VPC-Endpunktrichtlinie für die Amazon DocumentDB DocumentDB-API

Sie können Ihrem VPC-Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf die Amazon DocumentDB DocumentDB-API steuert. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuern des Zugriffs auf VPC-Endpunkte mithilfe von Endpunktrichtlinien im Amazon Virtual Private Cloud (AWS PrivateLink) -Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie für Amazon DocumentDB DocumentDB-API-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die Amazon DocumentDB DocumentDB-API. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten Amazon DocumentDB DocumentDB-API-Aktionen.

```
{
"Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "docdb:CreateDBInstance",
            "docdb:ModifyDBInstance",
            "docdb:CreateDBSnapshot"
         ],
         "Resource":"*"
      }
   ]
```

```
}
```

Beispiel: VPC-Endpunktrichtlinie, die jeglichen Zugriff von einem bestimmten Konto aus verweigert AWS

Die folgende VPC-Endpunktrichtlinie verweigert dem AWS Konto 123456789012 jeglichen Zugriff auf Ressourcen, die den Endpunkt verwenden. Die Richtlinie erlaubt alle Aktionen von anderen Konten.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
}
```

Bewährte Sicherheitsmethoden für Amazon DocumentDB

Aus Sicherheitsgründen müssen Sie AWS Identity and Access Management (IAM-) Konten verwenden, um den Zugriff auf Amazon DocumentDB DocumentDB-API-Operationen zu kontrollieren, insbesondere Operationen, die Amazon DocumentDB DocumentDB-Ressourcen erstellen, ändern oder löschen. Zu solchen Ressourcen gehören Cluster, Sicherheitsgruppen und Parametergruppen. Sie müssen IAM auch zur Steuerung von Aktionen verwenden, mit denen allgemeine Verwaltungsaktionen wie das Sichern und Wiederherstellen von Clustern ausgeführt werden. Verwenden Sie bei der Erstellung von IAM-Rollen das Prinzip der geringsten Rechte.

- Erzwingen Sie die geringste Berechtigung mit rollenbasierter Zugriffssteuerung.
- Weisen Sie jeder Person, die Amazon DocumentDB DocumentDB-Ressourcen verwaltet, ein individuelles IAM-Konto zu. Verwenden Sie den AWS-Konto Root-Benutzer nicht zur Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen. Erstellen Sie einen IAM-Benutzer für jede Person einschließlich Sie selbst.

- Gewähren Sie jedem -Benutzer nur den Mindestsatz an Berechtigungen, die für die Ausführung seiner Aufgaben erforderlich sind.
- Verwenden Sie IAM-Gruppen, um Berechtigungen für mehrere Benutzer effektiv zu verwalten. Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch. Weitere Informationen zu bewährten Methoden für IAM finden Sie unter Bewährte Methoden für IAM.
- Wechseln Sie regelmäßig die IAM-Anmeldeinformationen.
- Konfigurieren Sie AWS Secrets Manager so, dass die Secrets für Amazon DocumentDB automatisch rotiert werden. Weitere Informationen finden Sie unter Rotating Your AWS Secrets Manager Secrets und Rotating Secrets for Amazon DocumentDB im AWS Secrets Manager User Guide.
- Verwenden Sie Transport Layer Security (TLS) und die Verschlüsselung ruhender Daten, um Ihre Daten zu verschlüsseln.

Amazon DocumentDB DocumentDB-Ereignisse prüfen

Mit Amazon DocumentDB (mit MongoDB-Kompatibilität) können Sie Ereignisse überprüfen, die in Ihrem Cluster durchgeführt wurden. Beispiele für protokollierte Ereignisse sind erfolgreiche und fehlgeschlagene Authentifizierungsversuche, Drop-Ereignisse für Sammlungen in einer Datenbank oder das Erstellen eines Index. Standardmäßig ist die Prüfung in Amazon DocumentDB deaktiviert und erfordert, dass Sie sich für die Nutzung dieser Funktion anmelden.

Wenn die Prüfung aktiviert ist, zeichnet Amazon DocumentDB Ereignisse in Data Definition Language (DDL), Data Manipulation Language (DML), Authentifizierung, Autorisierung und Benutzerverwaltung in Amazon Logs auf. CloudWatch Wenn Auditing aktiviert ist, exportiert Amazon DocumentDB die Auditing-Datensätze (JSON-Dokumente) Ihres Clusters nach Amazon CloudWatch Logs. Sie können Amazon CloudWatch Logs verwenden, um Ihre Amazon DocumentDB-Prüfungsereignisse zu analysieren, zu überwachen und zu archivieren.

Amazon DocumentDB berechnet zwar keine zusätzlichen Kosten für die Aktivierung der Prüfung, Ihnen werden jedoch Standardtarife für die Nutzung von CloudWatch Logs berechnet. Informationen zu den Preisen für CloudWatch Logs finden Sie unter CloudWatch Amazon-Preise.

Die Amazon DocumentDB-Prüfungsfunktion unterscheidet sich deutlich von der Nutzung der Serviceressourcen, mit der überwacht wird. AWS CloudTrail CloudTrail zeichnet Operationen AWS Management Console auf, die mit AWS Command Line Interface (AWS CLI) oder mit Ressourcen wie Clustern, Instances, Parametergruppen und Snapshots ausgeführt werden. Die Überwachung

von Ressourcen mit CloudTrail ist standardmäßig aktiviert und kann nicht deaktiviert werden. Die Amazon DocumentDB-Prüfungsfunktion ist eine optionale Funktion. Sie zeichnet Operationen auf, die innerhalb Ihres Clusters für Objekte, wie z. B. Datenbanken, Sammlungen, Indizes und Benutzer ausgeführt werden.

Themen

- Unterstützte Ereignisse
- Auditing wird aktiviert
- Auditing wird deaktiviert
- Zugriff auf Ihre Prüfereignisse
- DML-Auditereignisse filtern

Unterstützte Ereignisse

Amazon DocumentDB DocumentDB-Auditing unterstützt die folgenden Ereigniskategorien:

- Data Definition Language (DDL) — umfasst Datenbankverwaltungsvorgänge, Verbindungen, Benutzerverwaltung und Autorisierung.
- Leseereignisse (DML-Lesevorgänge) in der Data Manipulation Language — umfasst die verschiedenen Aggregationsoperatoren, arithmetischen Operatoren, booleschen Operatoren `find()` und andere Leseabfrageoperatoren.
- Schreibereignisse (DML-Schreibvorgänge) in der Data Manipulation Language — beinhaltet Operatoren `insert(), update(), delete(), bulkWrite()`

Folgende Ereignistypen werden unterstützt:

Ereignistyp	Kategorie	Beschreibung
`authCheck`	Autorisierung	Ergebniscode 0: Erfolgreich
		Ergebniscode 13: Unbefugte Versuche, einen Vorgang auszuführen.

Ereignistyp	Kategorie	Beschreibung
`authenticate`	Verbindung	Erfolgreiche oder fehlgeschlagene Authentifizierungs versuche bei einer neuen Verbindung.
`auditConfigure`	DDL	Überprüfen Sie die Filterkonfiguration.
`createDatabase`	DDL	Erstellung einer neuen Datenbank.
`createCollection`	DDL	Erstellung einer neuen Sammlung innerhalb einer Datenbank.
`createIndex`	DDL	Erstellung eines neuen Index innerhalb einer Sammlung.
`dropCollection`	DDL	Löschen einer Sammlung in einer Datenbank.
`dropDatabase`	DDL	Löschen einer Datenbank.
`dropIndex`	DDL	Löschen eines Index innerhalb einer Sammlung.
`modifyChangeStreams`	DDL	Der Change-Stream wurde erstellt.

Ereignistyp	Kategorie	Beschreibung
`renameCollection`	DDL	Umbenennen einer Sammlung innerhalb einer Datenbank.
`createRole`	Rollenverwaltung	Eine Rolle erstellen.
`dropAllRolesFromDatabase`	Rollenverwaltung	Alle Rollen innerhalb einer Datenbank löschen
`dropRole`	Rollenverwaltung	Eine Rolle löschen.
`grantPrivilegesToRole`	Rollenverwaltung	Einer Rolle Privilegien gewähren
`grantRolesToRole`	Rollenverwaltung	Einer benutzerd efinierten Rolle Rollen zuweisen
`revokePrivilegesFromRole`	Rollenverwaltung	Rechte einer Rolle entziehen
`revokeRolesFromRole`	Rollenverwaltung	Sperren von Rollen aus einer benutzerd efinierten Rolle
`updateRole`	Rollenverwaltung	Eine Rolle aktualisi eren.
`createUser`	Benutzerverwaltung	Anlegen eines neuen Benutzers.
`dropAllUsersFromDatabase`	Benutzerverwaltung	Löschen aller Benutzer innerhalb einer Datenbank.

Ereignistyp	Kategorie	Beschreibung
`dropUser`	Benutzerverwaltung	Löschen eines bestehenden Benutzers.
`grantRolesToUser`	Benutzerverwaltung	Einem Benutzer Rollen zuweisen
`revokeRolesFromUser`	Benutzerverwaltung	Einem Benutzer Rollen entziehen
`updateUser`	UserManagement	Aktualisierung eines bestehenden Benutzers.
`insert`	DML schreiben	Fügt ein oder mehrere Dokumente in eine Sammlung ein.
`delete`	DML, schreiben	Löscht ein oder mehrere Dokumente aus einer Sammlung.
`update`	DML, schreiben	Ändert ein vorhanden es Dokument oder Dokumente in einer Sammlung.
`bulkWrite`	DML-Schreiben	Führt mehrere Schreiboperationen mit Steuerelementen für die Reihenfolge der Ausführung aus.
`setAuditConfig`	DML-Schreiben	Legen Sie einen neuen Filter für die DML-Überwachung fest.

Ereignistyp	Kategorie	Beschreibung
`count`	DML gelesen	Gibt die Anzahl der Dokumente zurück, die einer find () -Abfrage für die Sammlung oder Ansicht entsprechen würden.
`countDocuments`	DML gelesen	Gibt die Anzahl der Dokumente zurück, die der Abfrage für eine Sammlung oder Ansicht entsprechen.
`find`	DML gelesen	Wählt Dokumente in einer Sammlung oder Ansicht aus und bringt einen Cursor zu den ausgewählten Dokumenten zurück.
`getAuditConfig`	DML gelesen	Ruft den aktuellen Filter für die DML-Überwachung ab.
`findAndModify`	DML lesen und DML schreiben	Ändert ein einzelnes Dokument und gibt es zurück.
`findOneAndDelete`	DML lesen und DML schreiben	Löscht ein einzelnes Dokument auf der Grundlage der Filter- und Sortierkr iterien und gibt das gelöschte Dokument zurück.

Ereignistyp	Kategorie	Beschreibung
`findOneAndReplace`	DML lesen und DML schreiben	Ersetzt ein einzelnes Dokument auf der Grundlage des angegebenen Filters.
`findOneAndUpdate`	DML lesen und DML schreiben	Aktualisiert ein einzelnes Dokument auf der Grundlage der Filter- und Sortierkr iterien.
`aggregate`	DML lesen und DML schreiben	Unterstützt APIs in der Aggregationspipeli ne.
`distinct`	DML gelesen	Findet die unterschi edlichen Werte für ein bestimmtes Feld in einer einzelnen Sammlung oder Ansicht und gibt die Ergebnisse in einem Array zurück.

> **Note**
>
> Die Werte im Parameterfeld des DML-Ereignisdokuments haben eine Größenbeschränkung von 1 KB. Amazon DocumentDB kürzt den Wert, wenn er 1 KB überschreitet.

> **Note**
>
> TTL-Löschereignisse werden derzeit nicht geprüft.

Auditing wird aktiviert

Die Aktivierung des Prüfens für einen Cluster ist ein zweistufiger Prozess. Stellen Sie sicher, dass beide Schritte abgeschlossen sind, da andernfalls keine Auditprotokolle an CloudWatch Logs gesendet werden.

Schritt 1. Aktivieren Sie den Clusterparameter audit_logs

Um die Überwachung zu aktivieren, müssen Sie den `audit_logs` Parameter in der Parametergruppe ändern. `audit_logs`ist eine durch Kommas getrennte Liste von Ereignissen, die protokolliert werden sollen. Ereignisse müssen in Kleinbuchstaben angegeben werden und es darf kein Leerzeichen zwischen den Listenelementen sein.

Sie können die folgenden Werte für die Parametergruppe festlegen:

Wert	Beschreibung	
`ddl`	Diese Einstellung ermöglicht die Überwachung von DDL-Ereignissen wie CreateDatabase, DropDatabase, CreateCollection, DropCollection, CreateIndex, DropIndex, AuthCheck, authenticate, createUser, DropUser, User, updateUser und grantRolesTo revokeRolesFrom dropAllUsersFromDatabase	
`dml_read`	Diese Einstellung ermöglicht die	

Wert	Beschreibung	
	Prüfung von DML-Leseereignissen wie find, sort count, distinct, group, projecta, unwind, GeoNear, GeoIntersects, GeoWithin und anderen MongoDB-Leseabfrageoperatoren.	
`dml_write`	Wenn Sie diese Einstellung festlegen, wird die Überwachung von DML-Schreibereignissen wie insert (), update (), delete () und bulkWrite () aktiviert	
`all`	Wenn Sie diese Einstellung festlegen, wird die Überwachung Ihrer Datenbankereignisse wie Leseabfragen, Schreibabfragen, Datenbankaktionen und Administratoraktionen aktiviert.	
`none`	Wenn Sie diese Einstellung festlegen, wird die Überwachung deaktiviert	

Wert	Beschreibung	
`enabled` (veraltet)	Dies ist eine ältere Parametereinstellung, die 'ddl' entspricht. Diese Einstellung ermöglicht die Überwachung von DDL-Ereignissen wie CreateDatabase, DropDatabase, CreateCollection, DropCollection, CreateIndex, DropIndex, AuthCheck, authenticate, createUser, DropUser, User, updateUser und. grantRolesTo revokeRolesFrom dropAllUsersFromDatabase Es wird nicht empfohlen, diese Einstellung zu verwenden, da es sich um eine ältere Einstellung handelt.	

Wert	Beschreibung
`disabled` (veraltet)	Dies ist eine ältere Parametereinstellung, die „none“ entspricht. Wir empfehlen, diese Einstellung nicht zu verwenden, da es sich um eine ältere Einstellung handelt.

> **Note**
>
> Der Standardwert für den Clusterparameter audit_logs ist `none` (legacy "`disabled`„).

Sie können die oben genannten Werte auch in Kombinationen verwenden.

Wert	Beschreibung
`ddl, dml_read`	Wenn Sie diese Einstellung festlegen, wird die Überwachung von DDL-Ereignissen und DML-Lesee reignissen aktiviert.
`ddl, dml_write`	Wenn Sie diese Einstellung festlegen , wird die Überwachu ng von DDL-Ereig nissen und DML-Schreibvorgängen aktiviert
`dml_read, dml_write`	Wenn Sie diese Einstellung festlegen

Wert	Beschreibung	
	, wird die Überwachung für alle DML-Ereignisse aktiviert	

> **Note**
>
> Eine Standardparametergruppe kann nicht abgeändert werden.

Weitere Informationen finden Sie hier:

- Amazon DocumentDB-Cluster-Parametergruppen erstellen

  Nach dem Erstellen einer Parametergruppe ändern Sie diese, indem Sie den `audit_logs`-Parameterwert auf `all` ändern.
- Amazon DocumentDB-Cluster-Parametergruppen ändern

Schritt 2. Amazon CloudWatch Logs-Export aktivieren

Wenn der Wert des `audit_logs` Cluster-Parameters`enabled`,, oder `dml_write` ist `ddl`,`dml_read`, müssen Sie Amazon DocumentDB auch für den Export von Protokollen nach Amazon CloudWatch aktivieren. Wenn Sie einen dieser Schritte auslassen, werden keine Audit-Logs an gesendet. CloudWatch

Wenn Sie einen Cluster erstellen, einen Snapshot ausführen oder einen point-in-time-restore Snapshot wiederherstellen, können Sie CloudWatch Logs aktivieren, indem Sie die folgenden Schritte ausführen.

Using the AWS Management Console

Informationen zum Exportieren von Protokollen durch Amazon DocumentDB in die CloudWatch Konsole finden Sie in den folgenden Themen:

- Beim Erstellen eines Clusters — InErstellen eines Clusters und einer primären Instance mit dem AWS Management Console, siehe Cluster erstellen: Zusätzliche Konfigurationen (Schritt 5, Protokollexporte)

- Beim Ändern eines vorhandenen Clusters — Ändern eines Amazon DocumentDB-Clusters
- Bei der Durchführung einer Cluster-Snapshot-Wiederherstellung — Wiederherstellung aus einem Cluster-Snapshot
- Bei der Durchführung einer point-in-time Wiederherstellung — Zu einem bestimmten Zeitpunkt wiederherstellen

Using the AWS CLI

So aktivieren Sie Prüfungsprotokolle beim Erstellen eines neuen Clusters

Der folgende Code erstellt den Cluster `sample-cluster` und aktiviert CloudWatch Audit-Logs.

Example

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
    --db-cluster-identifier sample-cluster \
    --port 27017 \
    --engine docdb \
    --master-username master-username \
    --master-user-password password \
    --db-subnet-group-name default \
    --enable-cloudwatch-logs-exports audit
```

Für Windows:

```
aws docdb create-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --port 27017 ^
    --engine docdb ^
    --master-username master-username ^
    --master-user-password password ^
    --db-subnet-group-name default ^
    --enable-cloudwatch-logs-exports audit
```

So aktivieren Sie Prüfungsprotokolle beim Ändern eines vorhandenen Clusters

Der folgende Code modifiziert den Cluster `sample-cluster` und aktiviert CloudWatch Audit-Logs.

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Für Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"EnableLogTypes":["audit"]}'
```

Die Ausgabe dieser Operationen sieht in etwa wie folgt aus (JSON-Format).

```
{
    "DBCluster": {
        "HostedZoneId": "ZNKXH85TT8WVW",
        "StorageEncrypted": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "MasterUsername": "<user-name>",
        "BackupRetentionPeriod": 1,
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-cluster",
        "Status": "creating",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "DBSubnetGroup": "default",
        "DBClusterMembers": [],
```

```
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
        "DbClusterResourceId": "cluster-YOS52CUXGDTNKDQ7DH72I4LED4",
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "PreferredBackupWindow": "07:16-07:46",
        "DBClusterIdentifier": "sample-cluster"
    }
}
```

Auditing wird deaktiviert

Sie können die Überwachung deaktivieren, indem Sie den CloudWatch Protokollexport und den `audit_logs` Parameter deaktivieren.

Den Export von Protokollen deaktivieren CloudWatch

Sie können den Export von Auditprotokollen entweder mit dem AWS Management Console oder dem AWS CLI deaktivieren.

Using the AWS Management Console

Das folgende Verfahren verwendet die AWS Management Console , um den Export von Protokollen durch Amazon DocumentDB zu CloudWatch deaktivieren.

So deaktivieren Sie Prüfungsprotokolle

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster. Wählen Sie anschließend die Schaltfläche links neben dem Namen des Clusters aus, für den Sie das Exportieren von Protokollen deaktivieren möchten.
3. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.

4. Scrollen Sie nach unten zum Abschnitt Log exports (Protokollexporte) und wählen Sie Disabled (Deaktiviert) aus.
5. Klicken Sie auf Weiter.
6. Überprüfen Sie Ihre Änderungen. Wählen Sie anschließend den Zeitpunkt aus, an dem diese Änderung auf Ihren Cluster angewendet werden soll.
   - Apply during the next scheduled maintenance window (Anwendung während des nächsten geplanten Wartungsfensters)
   - Apply immediately (Sofort anwenden)
7. Wählen Sie Cluster bearbeiten aus.

Using the AWS CLI

Der folgende Code modifiziert den Cluster `sample-cluster` und deaktiviert Audit-Logs. CloudWatch

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

Für Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit"]}'
```

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

```
{
    "DBCluster": {
        "DBClusterParameterGroup": "default.docdb4.0",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "MasterUsername": "<user-name>",
        "Status": "available",
        "Engine": "docdb",
        "Port": 27017,
```

```
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "EarliestRestorableTime": "2019-02-13T16:35:50.387Z",
        "DBSubnetGroup": "default",
        "LatestRestorableTime": "2019-02-13T16:35:50.387Z",
        "DBClusterArn": "arn:aws:rds:us-east-1:900083794985:cluster:sample-
cluster2",
        "Endpoint": "sample-cluster2.cluster-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster2.cluster-ro-corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
        "BackupRetentionPeriod": 1,
        "EngineVersion": "4.0.0",
        "MultiAZ": false,
        "ClusterCreateTime": "2019-02-13T16:35:04.756Z",
        "DBClusterIdentifier": "sample-cluster2",
        "AssociatedRoles": [],
        "PreferredBackupWindow": "07:16-07:46",
        "DbClusterResourceId": "cluster-YOS52CUXGDTNKDQ7DH72I4LED4",
        "StorageEncrypted": false,
        "PreferredMaintenanceWindow": "wed:03:08-wed:03:38",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

Deaktivierung des Parameters audit_logs

Um den Parameter `audit_logs` für Ihren Cluster zu deaktivieren, können Sie den Cluster so ändern, dass er eine Parametergruppe mit dem Wert `disabled` für den Parameter `audit_logs` verwendet. Sie können auch den Wert des Parameters `audit_logs` in der Parametergruppe des Clusters in `disabled` ändern.

Weitere Informationen finden Sie unter den folgenden Themen:

- Ändern eines Amazon DocumentDB-Clusters
- Amazon DocumentDB-Cluster-Parametergruppen ändern

Zugriff auf Ihre Prüfereignisse

Gehen Sie wie folgt vor, um auf Ihre Prüfereignisse bei Amazon zuzugreifen CloudWatch.

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Stellen Sie sicher, dass Sie sich in derselben Region wie Ihr Amazon DocumentDB-Cluster befinden.
3. Wählen Sie im Navigationsbereich Protokolle aus.
4. Um die Prüfprotokolle für Ihren Cluster zu finden, suchen Sie in der Liste und wählen Sie **`/aws/docdb/`*`yourClusterName`*`/audit`** aus.

   Die Prüfereignisse für Ihre Instances sind unter dem jeweiligen Instance-Namen verfügbar.

DML-Auditereignisse filtern

Erste Schritte mit der DML-Auditfilterung

DML-Prüfereignisse können gefiltert werden, bevor sie an Amazon CloudWatch geschrieben werden. Um diese Funktion nutzen zu können, müssen das Auditprotokoll und die DML-Protokollierung aktiviert sein. Amazon DocumentDB unterstützt das Filtern nach `atypecommand`,`user`,`namespace`, und`auditAuthorizationSuccess`.

> **Note**
>
> DDL-Ereignisse werden nicht gefiltert.

Sie können die Überwachungsfilterung jederzeit aktivieren, indem Sie den Überwachungsfilter mithilfe der `auditAuthorizationSuccess` Parameter `setAuditConfigfilter`, und im `db.adminCommand( { command } )` Vorgang angeben:

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
```

```
      filter:
         {
            //filter conditions
         },
      auditAuthorizationSuccess: true | false
   }
)
```

Sie können die Auditfiltereinstellungen auch abrufen, indem Sie den folgenden Befehl ausführen:

```
db.admin.runCommand( { getAuditConfig: 1})
```

Sicherheitsanforderungen

Nur Datenbankbenutzer/Rollen mit Zugriffsrechten `auditConfigure` können die oben genannten Befehle ausführen, `admindb` wenn sie DML-Auditfilter setzen oder auflisten. Sie können entweder eine der integrierten Rollen von [`clusterAdmin`,`hostManager`,`root`] verwenden oder benutzerdefinierte Rollen mit Rechten erstellen. `auditConfigure` Im Folgenden finden Sie ein Beispiel für die Verwendung vorhandener Rollen mit dieser `auditConfigure` Berechtigung und ein Beispiel für benutzerdefinierte Rollen.

Benutzer mit integrierter Rolle:

```
use admin
db.createUser(
  {
    user: "myClusterAdmin",
    pwd: "password123",
    roles: [ { role: "clusterAdmin", db: "admin" } ]
  }
)
```

Benutzer mit benutzerdefinierten Rollen:

```
use admin
db.createRole(
   {
     role: "myRole",
     privileges: [
       { resource: { cluster: true }, actions: [ "auditConfigure" ] }
     ],
```

```
     roles: []
   }
)
db.createUser(
  {
    user: "myUser",
    pwd: "myPassword",
    roles: [ { role: "myRole", db: "admin" } ]
  }
)
```

Anwendungsfälle filtern

Beispiel: Ereignisse nach Befehlen filtern

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {
        "$and": [
         {
            "param.command":
               {
                  $in: [ "find","count", "insert", "delete", "update",
 "findandmodify" ]
               }
         }
         ]
      },
      auditAuthorizationSuccess: true
   }
)
```

Beispiel: Ereignisse nach Benutzernamen filtern

In diesem Beispiel wird nur der Benutzer „MyUser“ protokolliert:

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {
      "$and": [
         {
```

```
            "param.user":
               {
                  $in: [ "myUser" ]
               }
         }
         ]},
      auditAuthorizationSuccess: true})
```

Beispiel: Filterung nach **`atype`**

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {atype: "authCheck"},
      auditAuthorizationSuccess: true
   })
```

> **Note**
>
> Alle DML-Protokolle haben `authCheck` als`atype`. Nur DDL hat eine andere. `atype` Wenn Sie einen anderen Wert als `authCheck` in den eingeben`filter`, wird kein DML-Login erzeugt. CloudWatch

Beispiel: Filterung mithilfe mehrerer Filter, die durch Operatoren miteinander verbunden sind

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {
      "$and": [
         {
            "param.command":
               {
                  $in: [ "find","count", "insert", "delete", "update",
 "findandmodify" ]
               }
         }
         ],
       "$nor": [
         {
```

```
            "param.command":
               {
                  $in: ["count", "insert", "delete", "update", "findandmodify" ]
               }
         }]
       },
      auditAuthorizationSuccess: true})
```

> **Note**
>
> Auf der obersten Ebene `$nor` werden nur `$and``$or`, und unterstützt. Alle anderen Operatoren werden nicht unterstützt und führen zu einem Fehler.

Beispiel: Filterung nach Ereignissen nach **`auditAuthorizationSuccess`**

In diesem Filter werden alle Befehle, die die Autorisierung erfolgreich bestanden haben, nicht protokolliert:

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {},
      auditAuthorizationSuccess: false
   }
)
```

Beispiel: Filtern mit **`$in`** und **`$nin`** -Bedingungen

Wenn `$in` sowohl in als auch verwendet wird`$nin`, wird der Befehl nicht protokolliert, da zwischen den Bedingungen ein implizites „und“ steht. In diesem Beispiel blockiert Regex den `find` Befehl, sodass nichts protokolliert wird:

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {
      "$and": [
         {
            atype: "authCheck",
            "param.command":
```

```
               {
                  $in: [ "find", "insert", "delete", "update", "findandmodify" ],
                  $nin: ["count", "insert", "delete", "update", "findandmodify" ],
                  $not: /^^find.*/
               }
         },
         ],
       "$or": [
         {
            "param.command":
               {
                  $nin: ["count", "insert", "delete", "update", "findandmodify" ]
               }
         }]
       },
      auditAuthorizationSuccess: true})
```

Beispiel: Filtern nach **`namespace`**

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {
      "$and": [
         {
            "param.ns":
               {
                  $in: [ "test.foo" ]
               }
         }
         ]},
      auditAuthorizationSuccess: true})
```

Beispiel: Zurücksetzen auf den Standardfilter

Das Zurücksetzen auf den Standardwert bedeutet, dass jedes DML-Auditereignis protokolliert wird. Führen Sie den folgenden Befehl aus, um die Filterung auf den Standardwert zurückzusetzen:

```
db.admin.runCommand(
   {
      setAuditConfig: 1,
      filter: {},
      auditAuthorizationSuccess: true
```

```
    }
)
```

Sichern und Wiederherstellen in Amazon DocumentDB

Amazon DocumentDB (mit MongoDB-Kompatibilität) sichert Ihre Daten kontinuierlich 1—35 Tage lang auf Amazon Simple Storage Service (Amazon S3), sodass Sie sie an jedem beliebigen Punkt innerhalb der Aufbewahrungsfrist für Backups schnell wiederherstellen können. Amazon DocumentDB erstellt im Rahmen dieses kontinuierlichen Backup-Prozesses auch automatische Schnappschüsse Ihrer Daten.

> **Note**
>
> Dies sind vom Service verwaltete Amazon S3 S3-Buckets, und Sie haben keinen Zugriff auf die Sicherungsdateien. Wenn Sie Ihre eigenen Backups kontrollieren möchten, folgen Sie den Anweisungen zum Dumping, Wiederherstellen, Importieren und Exportieren von Daten.

Sie können Backup-Daten auch über die Aufbewahrungsfrist hinaus aufbewahren, indem Sie einen manuellen Snapshot der Daten Ihres Clusters erstellen. Der Backup-Prozess hat keinen Einfluss auf die Leistung Ihres Clusters.

In diesem Abschnitt werden die Anwendungsfälle für die Backup-Funktionen in Amazon DocumentDB beschrieben und es wird gezeigt, wie Sie Backups für Ihre Amazon DocumentDB-Cluster verwalten.

Themen

- Sichern und Wiederherstellen: Konzepte
- Grundlegendes zur Sicherungsspeicher-Nutzung in
- Daten speichern, wiederherstellen, importieren und exportieren
- Überlegungen zu Cluster-Snapshots
- Vergleich von automatischen und manuellen Snapshots
- Einen manuellen Cluster-Snapshot erstellen
- Amazon DocumentDB-Cluster-Snapshots kopieren
- Amazon DocumentDB-Cluster-Snapshots teilen
- Wiederherstellung aus einem Cluster-Snapshot
- Zu einem bestimmten Zeitpunkt wiederherstellen
- Löschen eines Cluster-Snapshots

Sichern und Wiederherstellen: Konzepte

Substantiv	Beschreibung	APIs (Verben)

Aufbewahrungszeitraum für Backups	Ein Zeitraum zwischen 1 und 35 Tagen, für den Sie eine point-in-time Wiederherstellung durchführen können.	`create-db-cluster`
		`modify-db-cluster`
		`restore-db-cluster-to-point-in-time`

Amazon DocumentDB DocumentDB-Speichervolumen	Hochverfügbares und sehr robustes Speichervolume, das Daten auf sechs Arten in drei Availability Zones repliziert. Ein Amazon DocumentDB-Cluster ist	`create-db-cluster` `delete-db-cluster`

Substantiv	Beschreibung	APIs (Verben)
	unabhängig von der Anzahl der Instances im Cluster äußerst robust.	

Backup-Fenster	Zeitspanne an dem Tag, an dem automatische Snapshots gemacht werden.	`create-db-cluster`
		`describe-db-cluster`
		`modify-db-cluster`

Substantiv	Beschreibung	APIs (Verben)

Automatischer Snapshot	Tägliche Snapshots, bei denen es sich um vollständige Backups des Clusters handelt, die automatisch durch den kontinuierlichen Backup-Prozess in Amazon DocumentDB erstellt werden.	`restore-db-cluster-from-snapshot`
		
		`describe-db-cluster-snapshot-attributes`
		
		`describe-db-cluster-snapshots`

Substantiv	Beschreibung	APIs (Verben)

Manueller Snapshot	Snapshots, die Sie manuell erstellen, um Vollsicherungen eines Clusters über den Sicherungszeitraum hinaus aufzubewahren.	`create-db-cluster-snapshot`
		`copy-db-cluster-snapshot`
		`delete-db-cluster-snapshot`
		`describe-db-cluster-snapshot-attributes`
		`describe-db-cluster-snapshots`
		`modify-db-cluster-snapshot-attribute`

Grundlegendes zur Sicherungsspeicher-Nutzung in

Der Amazon DocumentDB-Backup-Speicher besteht aus kontinuierlichen Backups innerhalb des Aufbewahrungszeitraums für Backups und manuellen Snapshots außerhalb des Aufbewahrungszeitraums. Wenn Sie Ihren Sicherungsspeicher kontrollieren möchten, können Sie das Aufbewahrungsintervall für Sicherungen verringern, alte (nicht mehr benötigte) manuelle Snapshots entfernen oder beides. Allgemeine Informationen zu Amazon DocumentDB-Backups finden Sie unterSichern und Wiederherstellen in Amazon DocumentDB. Preisinformationen zum Amazon DocumentDB-Backup-Speicher finden Sie unter Amazon DocumentDB DocumentDB-Preise.

Wenn Sie Ihre Kosten kontrollieren möchten, können Sie die Menge des Speichers überwachen, der nach Ablauf des Aufbewahrungszeitraums noch von kontinuierlichen Sicherungen und manuellen Snapshots belegt wird. Anschließend können Sie den Aufbewahrungszeitraum für Backups verringern und manuelle Snapshots entfernen, wenn sie nicht mehr benötigt werden.

Sie können die CloudWatch Amazon-Metriken`TotalBackupStorageBilled`,`SnapshotStorageUsed`, wie folgt verwenden,

`BackupRetentionPeriodStorageUsed` um den Speicherverbrauch Ihrer Amazon DocumentDB-Backups zu überprüfen und zu überwachen:

- `BackupRetentionPeriodStorageUsed` stellt den Umfang des Backup-Speichers zum Speichern von kontinuierlichen Backups zur aktuellen Zeit dar. Dieser Metrikwert hängt von der Größe des Cluster-Volumens und der Menge der von Ihnen während des Aufbewahrungszeitraums vorgenommenen Änderungen ab. Für Fakturierungszwecke übersteigt die Metrik die kumulative Cluster-Volume-Größe während des Aufbewahrungszeitraums nicht. Beispiel: Wenn Ihr Cluster 100 GiB groß ist und Ihr Aufbewahrungszeitraum zwei Tage beträgt, hat `BackRetentionPeriodStorageUsed` einen Maximalwert von 200 GiB (100 GiB+100 GiB).

- `SnapshotStorageUsed` stellt den Umfang des Backup-Speichers zum Speichern von manuellen Snapshots außerhalb des Aufbewahrungszeitraums für Backups dar. Innerhalb des Aufbewahrungszeitraums manuell aufgenommene Snapshots zählen nicht zu Ihrem Backup-Speicher. Ebenso werden automatische Snapshots nicht zu Ihrem Backup-Speicher gezählt. Die Größe einzelner Snapshots entspricht der Größe des Cluster-Volumes zum Zeitpunkt der Snapshot-Erstellung. Der `SnapshotStorageUsed`-Wert hängt von der Anzahl der Snapshots ab, die Sie aufbewahren, und von der Größe der einzelnen Snapshots. Beispiel: Angenommen, Sie haben einen Snapshot außerhalb des Aufbewahrungszeitraums und die Cluster-Volume-Größe betrug 100 GiB, als der Snapshot erstellt wurde. Dann beträgt die `SnapshotStorageUsed`-Menge 100 GiB.

- `TotalBackupStorageBilled` stellt die Summe von `BackupRetentionPeriodStorageUsed` und `SnapshotStorageUsed` abzüglich des freien Backup-Speichers gleich der Größe des Cluster-Volumes für einen Tag dar. Wenn Ihre Clustergröße beispielsweise 100 GiB beträgt, haben Sie einen Aufbewahrungstag und Sie haben einen Snapshot außerhalb des Aufbewahrungszeitraums, dieser `TotalBackupStorageBilled` beträgt 100 GiB (100 GiB + 100 GiB — 100 GiB).

- Diese Metriken werden unabhängig für jeden Amazon DocumentDB-Cluster berechnet.

Sie können Ihre Amazon DocumentDB-Cluster überwachen und Berichte mithilfe von CloudWatch Metriken über die CloudWatch Konsole erstellen. Weitere Informationen zur Verwendung von CloudWatch Metriken finden Sie unterÜberwachung von Amazon DocumentDB.

Daten speichern, wiederherstellen, importieren und exportieren

Sie können die `mongoimport` Dienstprogramme`mongodump`,`mongorestore`, und verwenden`mongoexport`, um Daten in und aus Ihrem Amazon DocumentDB-Cluster zu verschieben. In diesem Abschnitt wird der Zweck der einzelnen Tools und Konfigurationen beschrieben, damit Sie die Leistung verbessern können.

Themen

- mongodump
- mongorestore
- mongoexport
- mongoimport
- Tutorial

`mongodump`

Das Dienstprogramm `mongodump` erstellt eine binäre (BSON)-Sicherung einer MongoDB-Datenbank. Das `mongodump` Tool ist aufgrund der Größeneffizienz, die durch das Speichern der Daten in einem Binärformat erzielt wird, die bevorzugte Methode zum Dumping von Daten aus Ihrer MongoDB-Quellbereitstellung, wenn Sie sie in Ihrem Amazon DocumentDB-Cluster wiederherstellen möchten.

Abhängig von den Ressourcen, die auf der Instanz oder dem Computer verfügbar sind, den Sie für die Ausführung des Befehls verwenden, können Sie Ihren Vorgang beschleunigen, `mongodump` indem Sie die Anzahl der parallel Sammlungen erhöhen, die von der Standardeinstellung 1 abgerufen werden, indem Sie die `--numParallelCollections` Option verwenden. Eine gute Faustregel lautet, mit einem Worker pro vCPU auf der primären Instance Ihres Amazon DocumentDB-Clusters zu beginnen.

> **Note**
>
> Wir empfehlen MongoDB Database Tools bis einschließlich Version 100.6.1 für Amazon DocumentDB. Sie können hier auf die Downloads der MongoDB Database Tools zugreifen.

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des mongodump Dienstprogramms im Amazon DocumentDB-Cluster,`sample-cluster`.

```
mongodump --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --out=sample-output-file \
    --numParallelCollections 4  \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

`mongorestore`

Mit dem Dienstprogramm `mongorestore` können Sie eine Binärsicherung (BSON) einer Datenbank wiederherstellen, die mit dem Dienstprogramm `mongodump` erstellt wurde. Sie können die Wiederherstellungsleistung verbessern, indem Sie die Anzahl der Worker für jede Sammlung während der Wiederherstellung mit der Option `--numInsertionWorkersPerCollection` erhöhen. (Der Standardwert ist 1.) Eine gute Faustregel lautet, mit einem Worker pro vCPU auf der primären Instance Ihres Amazon DocumentDB-Clusters zu beginnen.

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des `mongorestore` Dienstprogramms im Amazon DocumentDB-Cluster,`sample-cluster`.

```
mongorestore --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem <fileToBeRestored>
```

`mongoexport`

Das `mongoexport` Tool exportiert Daten in Amazon DocumentDB in die Dateiformate JSON, CSV oder TSV. Das `mongoexport`-Tool ist die bevorzugte Methode für den Export von Daten, die für Menschen oder Maschinen lesbar sein müssen.

> **Note**
>
> `mongoexport` unterstützt parallele Exporte nicht direkt. Es ist jedoch möglich, die Leistung zu steigern, indem mehrere `mongoexport`-Aufgaben gleichzeitig für verschiedene Sammlungen ausgeführt werden.

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des `mongoexport` Tools im Amazon DocumentDB-Cluster,`sample-cluster`.

```
mongoexport --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --out=sample-output-file \
    --username=sample-user \
    --password=abc0123 \
    --sslCAFile global-bundle.pem
```

`mongoimport`

Das `mongoimport` Tool importiert den Inhalt von JSON-, CSV- oder TSV-Dateien in einen Amazon DocumentDB-Cluster. Mithilfe des Parameters `--numInsertionWorkers` können Sie den Import parallelisieren und beschleunigen. (Der Standardwert ist 1.)

Beispielverwendung

Im Folgenden finden Sie ein Beispiel für die Verwendung des `mongoimport` Tools im Amazon DocumentDB-Cluster,`sample-cluster`.

```
mongoimport --ssl \
    --host="sample-cluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=sample-collection \
    --db=sample-database \
    --file=<yourFile> \
    --numInsertionWorkers 4 \
    --username=sample-user \
    --password=abc0123 \
```

```
    --sslCAFile global-bundle.pem
```

Tutorial

Das folgende Tutorial beschreibt, wie Sie die `mongoimport` Dienstprogramme`mongodump`,`mongorestore`, und verwenden`mongoexport`, um Daten in und aus einem Amazon DocumentDB-Cluster zu verschieben.

1. Voraussetzungen — Bevor Sie beginnen, stellen Sie sicher, dass Ihr Amazon DocumentDB-Cluster bereitgestellt ist und dass Sie Zugriff auf eine EC2 Amazon-Instance in derselben VPC wie Ihr Cluster haben. Weitere Informationen finden Sie unter Stellen Sie über Amazon eine Verbindung her EC2.

   Um die Mongo Utility-Tools verwenden zu können, muss das mongodb-org-tools Paket wie folgt in Ihrer EC2 Instance installiert sein.

   ```
   sudo yum install mongodb-org-tools-4.0.18
   ```

   Da Amazon DocumentDB standardmäßig die Transport Layer Security (TLS) -Verschlüsselung verwendet, müssen Sie auch die Datei der Amazon RDS-Zertifizierungsstelle (CA) herunterladen, um die Mongo-Shell für die Verbindung zu verwenden. Gehen Sie dazu wie folgt vor.

   ```
   wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
   ```

2. Beispieldaten herunterladen — Für dieses Tutorial laden Sie einige Beispieldaten herunter, die Informationen über Restaurants enthalten.

   ```
   wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/
   datasets/restaurant.json
   ```

3. Importieren Sie die Beispieldaten in Amazon DocumentDB — Da die Daten in einem logischen JSON-Format vorliegen, verwenden Sie das `mongoimport` Hilfsprogramm, um die Daten in Ihren Amazon DocumentDB-Cluster zu importieren.

   ```
   mongoimport --ssl \
       --host="tutorialCluster.amazonaws.com:27017" \
       --collection=restaurants \
       --db=business \
   ```

```
    --file=restaurant.json \
    --numInsertionWorkers 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

4. Daten ausgeben mit **`mongodump`** — Da Sie nun Daten in Ihrem Amazon DocumentDB-Cluster haben, können Sie mit dem Hilfsprogramm einen binären Speicherauszug dieser Daten erstellen. `mongodump`

```
mongodump --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017"\
    --collection=restaurants \
    --db=business \
    --out=restaurantDump.bson \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

5. **`restaurants`**Sammlung löschen — Bevor Sie die `restaurants` Sammlung in der `business` Datenbank wiederherstellen, müssen Sie zunächst die Sammlung löschen, die bereits in dieser Datenbank vorhanden ist, und zwar wie folgt.

```
use business
```

```
db.restaurants.drop()
```

6. Daten wiederherstellen mit **`mongorestore`** — Mit dem binären Speicherauszug der Daten aus Schritt 3 können Sie jetzt das `mongorestore` Hilfsprogramm verwenden, um Ihre Daten in Ihrem Amazon DocumentDB-Cluster wiederherzustellen.

```
mongorestore --ssl \
    --host="tutorialCluster.us-east-1.docdb.amazonaws.com:27017" \
    --numParallelCollections 4 \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem restaurantDump.bson
```

7. Daten exportieren mit **`mongoexport`** — Um das Tutorial abzuschließen, exportieren Sie die Daten aus Ihrem Cluster im Format einer JSON-Datei, die sich nicht von der Datei unterscheidet, die Sie in Schritt 1 importiert haben.

```
mongoexport --ssl \
    --host="tutorialCluster.node.us-east-1.docdb.amazonaws.com:27017" \
    --collection=restaurants \
    --db=business \
    --out=restaurant2.json \
    --username=<yourUsername> \
    --password=<yourPassword> \
    --sslCAFile global-bundle.pem
```

8. Validierung — Mit den folgenden Befehlen können Sie überprüfen, ob die Ausgabe von Schritt 5 dasselbe Ergebnis wie Schritt 1 liefert.

```
wc -l restaurant.json
```

Ausgabe dieses Befehls:

```
2548 restaurant.json
```

```
wc -l restaurant2.json
```

Ausgabe dieses Befehls:

```
2548 restaurant2.json
```

Überlegungen zu Cluster-Snapshots

Amazon DocumentDB erstellt täglich automatische Snapshots Ihres Clusters während des Backup-Fensters Ihres Clusters. Amazon DocumentDB speichert die automatischen Snapshots Ihres Clusters entsprechend dem von Ihnen angegebenen Aufbewahrungszeitraum für Backups. Bei Bedarf können Sie Ihren Cluster zu einem beliebigen Zeitpunkt während der Aufbewahrungsdauer des Backups wiederherstellen. Automatische Snapshots treten nicht auf, während ein Kopiervorgang in derselben Region für denselben Cluster ausgeführt wird.

Themen

- Backup-Speicher
- Backup-Fenster
- Aufbewahrungszeitraum für Backups
- Kopieren Sie die Cluster-Snapshot-

Zusätzlich zu den automatischen Cluster-Snapshots können Sie auch manuell einen Cluster-Snapshot erstellen. Sie können sowohl automatische als auch manuelle Snapshots kopieren. Weitere Informationen erhalten Sie unter Einen manuellen Cluster-Snapshot erstellen und Amazon DocumentDB-Cluster-Snapshots kopieren.

> **Note**
>
> Ihr Cluster muss sich im Status verfügbar befinden, damit ein automatischer Snapshot erstellt werden kann.
> Sie können keinen automatisierten Amazon DocumentDB-Cluster-Snapshot teilen. Als Problemumgehung können Sie einen manuellen Snapshot erstellen, indem Sie den automatisierten Snapshot kopieren und dann diese Kopie freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Amazon DocumentDB-Cluster-Snapshots kopieren. Weitere Informationen zum Wiederherstellen eines Clusters aus einem Snapshot finden Sie unter Wiederherstellung aus einem Cluster-Snapshot.

Backup-Speicher

Ihr Amazon DocumentDB-Backup-Speicher AWS-Region besteht aus dem Backup-Speicher, der für Ihren Backup-Aufbewahrungszeitraum benötigt wird, einschließlich automatischer und manueller Cluster-Snapshots in dieser Region. Der standardmäßige Aufbewahrungszeitraum für Backups beträgt 1 Tag. Weitere Informationen zu den Preisen für Backup-Speicher finden Sie unter Amazon DocumentDB DocumentDB-Preise.

Wenn Sie einen Cluster löschen, werden alle seine automatischen Snapshots gelöscht und können nicht wiederhergestellt werden. Manuelle Snapshots werden jedoch nicht gelöscht, wenn Sie einen Cluster löschen. Wenn Sie Amazon DocumentDB einen endgültigen Snapshot (manueller Snapshot) erstellen lassen möchten, bevor Ihr Cluster gelöscht wird, können Sie den endgültigen Snapshot verwenden, um Ihren Cluster wiederherzustellen.

Weitere Informationen zu Snapshots und Speicher finden Sie unter Grundlegendes zur Sicherungsspeicher-Nutzung in .

Backup-Fenster

Automatische Snapshots erfolgen täglich während des bevorzugten Backup-Fensters. Wenn der Snapshot mehr Zeit benötigt, als dem Backup-Fenster zugewiesen wurde, wird der Backup-Prozess bis zum Ende fortgesetzt, auch wenn das Backup-Fenster beendet ist. Das Backup-Fenster kann sich nicht mit dem wöchentlichen Wartungsfenster für den Cluster überschneiden.

Wenn Sie bei der Erstellung des Clusters kein bevorzugtes Backup-Fenster angeben, weist Amazon DocumentDB ein standardmäßiges 30-minütiges Backup-Fenster zu. Dieses Fenster wird nach dem Zufallsprinzip aus einem 8-stündigen Zeitblock ausgewählt, der mit der Region Ihres Clusters verknüpft ist. Sie können Ihr bevorzugtes Backup-Fenster ändern, indem Sie den Cluster ändern. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Name der Region	Region	UTC-Zeitblock
USA Ost (Ohio)	us-east-2	03:00-11:00
USA Ost (Nord-Virginia)	us-east-1	03:00-11:00
USA West (Oregon)	us-west-2	06:00-14:00
Afrika (Kapstadt)	af-south-1	03:00 — 11:00
Asien-Pazifik (Hongkong)	ap-east-1	06:00-14:00
Asien-Pazifik (Hyderabad)	ap-south-2	06:30 — 14:30
Asien-Pazifik (Mumbai)	ap-south-1	06:00-14:00
Asien-Pazifik (Seoul)	ap-northeast-2	13:00-21:00
Asien-Pazifik (Singapur)	ap-southeast-1	14:00-22:00
Asien-Pazifik (Sydney)	ap-southeast-2	12:00-20:00
Asien-Pazifik (Tokio)	ap-northeast-1	13:00-21:00
Kanada (Zentral)	ca-central-1	03:00-11:00

Name der Region	Region	UTC-Zeitblock
China (Beijing)	cn-north-1	06:00-14:00
China (Ningxia)	cn-northwest-1	06:00-14:00
Europa (Frankfurt)	eu-central-1	21:00-05:00
Europa (Irland)	eu-west-1	22:00-06:00
Europa (London)	eu-west-2	22:00-06:00
Europa (Mailand)	eu-south-1	02:00-10:00
Europa (Paris)	eu-west-3	23:59-07:29
Europa (Spanien)	eu-south-2	02:00 — 10:00
Europa (Stockholm)	eu-north-1	04:00 — 12:00
Naher Osten (VAE)	me-central-1	05:00 — 13:00
Südamerika (São Paulo)	sa-east-1	00:00-08:00
Israel (Tel Aviv)	il-central-1	04:00-12:00
AWS GovCloud (US-Ost)	us-gov-east-1	17:00-01:00
AWS GovCloud (US-West)	us-gov-west-1	06:00-14:00

Aufbewahrungszeitraum für Backups

Die Aufbewahrungsfrist für Backups gibt an, wie viele Tage ein automatisches Backup aufbewahrt wird, bevor es automatisch gelöscht wird. Amazon DocumentDB unterstützt eine Aufbewahrungsdauer von Backups von 1—35 Tagen.

Sie können die Aufbewahrungsdauer des Backups beim Anlegen eines Clusters festlegen. Wenn Sie die Aufbewahrungsdauer für das Backup nicht explizit festlegen, wird Ihrem Cluster die standardmäßige Aufbewahrungsdauer für das Backup von 1 Tag zugewiesen. Nachdem Sie einen Cluster erstellt haben, können Sie den Aufbewahrungszeitraum für Backups ändern, indem Sie den

Cluster entweder mit AWS Management Console oder dem ändern. AWS CLI Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Kopieren Sie die Cluster-Snapshot-

Die Cluster- und Snapshot-Verschlüsselung basiert auf einem KMS-Verschlüsselungsschlüssel. Die KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN), der KMS-Schlüsselbezeichner oder der KMS-Schlüsselalias für den KMS-Verschlüsselungsschlüssel.

Es gelten die folgenden Richtlinien und Einschränkungen:

- Bei der Erstellung eines Snapshots wird die Verschlüsselung aus dem Cluster abgeleitet. Wenn der Cluster verschlüsselt ist, wird der Snapshot dieses Clusters mit demselben KMS-Schlüssel verschlüsselt. Wenn der Cluster nicht verschlüsselt ist, ist der Snapshot nicht verschlüsselt.
- Wenn Sie einen verschlüsselten Cluster-Snapshot von Ihrem Amazon Web Services Services-Konto kopieren, können Sie einen Wert angeben, `KmsKeyId` um die Kopie mit einem neuen KMS-Verschlüsselungsschlüssel zu verschlüsseln. Wenn Sie keinen Wert für angeben`KmsKeyId`, wird die Kopie des Cluster-Snapshots mit demselben KMS-Schlüssel verschlüsselt wie der Quell-Cluster-Snapshot.
- Wenn Sie einen verschlüsselten Cluster-Snapshot kopieren, der von einem anderen Amazon Web Services Services-Konto gemeinsam genutzt wird, müssen Sie einen Wert für angeben`KmsKeyId`.
- Um einen verschlüsselten Cluster-Snapshot in eine andere Amazon Web Services Services-Region `KmsKeyId` zu kopieren, legen Sie die KMS-Schlüssel-ID fest, mit der Sie die Kopie des Cluster-Snapshots in der Zielregion verschlüsseln möchten. KMS-Verschlüsselungsschlüssel sind spezifisch für die Amazon Web Services Services-Region, in der sie erstellt wurden, und Sie können keine Verschlüsselungsschlüssel aus einer Amazon Web Services Services-Region in einer anderen Amazon Web Services Services-Region verwenden.
- Wenn Sie einen unverschlüsselten Cluster-Snapshot kopieren und einen Wert für den `KmsKeyId` Parameter angeben, wird ein Fehler zurückgegeben.

Vergleich von automatischen und manuellen Snapshots

Im Folgenden sind die wichtigsten Funktionen der automatischen und manuellen Snapshots von Amazon DocumentDB (mit MongoDB-Kompatibilität) aufgeführt.

Automatische Amazon DocumentDB-Snapshots haben die folgenden Hauptfunktionen:

- Automatische Snapshot-Benennung — Automatische Snapshot-Namen folgen dem Muster `rds:<cluster-name>-yyyy-mm-dd-hh-mm` und geben das `yyyy-mm-dd-hh-mm` Datum und die Uhrzeit der Snapshot-Erstellung an.
- Automatisch nach einem Zeitplan erstellt — Wenn Sie einen Cluster erstellen oder ändern, können Sie den Aufbewahrungszeitraum für Backups auf einen ganzzahligen Wert zwischen 1 und 35 Tagen festlegen. Standardmäßig haben neue Cluster eine Aufbewahrungsdauer von 1 Tag. Die Aufbewahrungsdauer für Backups definiert die Anzahl der Tage, die automatische Snapshots aufbewahrt werden, bevor sie automatisch gelöscht werden. Sie können automatische Backups auf Amazon DocumentDB-Clustern nicht deaktivieren.

  Zusätzlich zur Einstellung der Aufbewahrungsdauer für das Backup legen Sie auch das Backup-Fenster fest (die Tageszeit, zu der automatische Snapshots erstellt werden).
- Automatische Snapshots löschen — Automatische Snapshots werden gelöscht, wenn Sie den Cluster des automatischen Snapshots löschen. Sie können einen automatischen Snapshot nicht manuell löschen.
- Inkrementell — Während der Aufbewahrungszeit der Backups werden Datenbankaktualisierungen aufgezeichnet, sodass eine inkrementelle Aufzeichnung der Änderungen erfolgt.
- Wiederherstellung aus einem automatischen Snapshot — Sie können die Wiederherstellung aus einem automatischen Snapshot mit dem AWS Management Console oder dem durchführen. AWS CLI Wenn Sie mithilfe von aus einem Snapshot wiederherstellen AWS CLI, müssen Sie Instanzen separat hinzufügen, sobald der Cluster verfügbar ist.
- Teilen — Sie können einen automatisierten Amazon DocumentDB-Cluster-Snapshot nicht teilen. Als Problemumgehung können Sie einen manuellen Snapshot erstellen, indem Sie den automatisierten Snapshot kopieren und dann diese Kopie freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Amazon DocumentDB-Cluster-Snapshots kopieren. Weitere Informationen zum Wiederherstellen eines Clusters aus einem Snapshot finden Sie unter Wiederherstellung aus einem Cluster-Snapshot.
- Sie können von jedem beliebigen Zeitpunkt innerhalb der Aufbewahrungsfrist für Backups aus wiederherstellen — Da Datenbankaktualisierungen inkrementell aufgezeichnet werden, können Sie Ihren Cluster zu jedem beliebigen Zeitpunkt innerhalb der Aufbewahrungsfrist für Backups wiederherstellen.

Wenn Sie aus einem automatischen Snapshot oder aus einer point-in-time Wiederherstellung mithilfe von wiederherstellen AWS CLI, müssen Sie Instanzen separat hinzufügen, sobald der Cluster verfügbar ist.

Manuelle Amazon DocumentDB-Snapshots haben die folgenden Hauptfunktionen:

- Auf Anfrage erstellt — Manuelle Amazon DocumentDB-Snapshots werden bei Bedarf mit der Amazon DocumentDB-Managementkonsole oder erstellt. AWS CLI
- Löschen eines manuellen Snapshots — Ein manueller Snapshot wird nur gelöscht, wenn Sie ihn explizit mit der Amazon DocumentDB DocumentDB-Konsole oder AWS CLI löschen. Ein manueller Snapshot wird nicht gelöscht, wenn Sie seinen Cluster löschen.
- Vollständige Backups — Wenn ein manueller Snapshot erstellt wird, wird eine vollständige Sicherung der Daten Ihres Clusters erstellt und gespeichert.
- Manuelle Snapshot-Benennung — Sie geben den Namen des manuellen Snapshots an. Amazon DocumentDB fügt dem Namen keinen `datetime` Stempel hinzu. Sie müssen diese Informationen also hinzufügen, wenn Sie möchten, dass sie in den Namen aufgenommen werden.
- Wiederherstellung aus einem manuellen Snapshot — Sie können die Wiederherstellung aus einem manuellen Snapshot mit der Konsole oder dem durchführen. AWS CLI Wenn Sie die Wiederherstellung aus einem Snapshot mithilfe von AWS CLI durchführen, müssen Sie Instanzen separat hinzufügen, sobald der Cluster verfügbar ist.
- Servicekontingente — Sie sind auf maximal 100 manuelle Snapshots pro Person AWS-Region beschränkt.
- Teilen — Sie können manuelle Cluster-Snapshots teilen, die von autorisierten Personen kopiert werden können. AWS-Konten Sie können verschlüsselte oder unverschlüsselte manuelle Snapshots freigeben. Weitere Informationen zum Kopieren eines Snapshots finden Sie unter Amazon DocumentDB-Cluster-Snapshots kopieren.
- Sie stellen den Zeitpunkt wieder her, zu dem der manuelle Snapshot erstellt wurde — Wenn Sie eine Wiederherstellung von einem manuellen Snapshot durchführen, stellen Sie den Zeitpunkt wieder her, zu dem der manuelle Snapshot erstellt wurde.

  Wenn Sie die Wiederherstellung aus einem Snapshot mithilfe von AWS CLI durchführen, müssen Sie Instanzen separat hinzufügen, sobald der Cluster verfügbar ist.

Einen manuellen Cluster-Snapshot erstellen

Sie können einen manuellen Snapshot entweder mit AWS Management Console oder AWS CLI erstellen. Die Zeit, die für die Erstellung eines Snapshots benötigt wird, hängt von der Größe Ihrer Datenbanken ab. Wenn Sie einen Snapshot erstellen, müssen Sie Folgendes tun:

1. Identifizieren Sie den zu sichernden Cluster.
2. Geben Sie Ihrem Snapshot einen Namen. Dies ermöglicht es Ihnen, ihn später wiederherzustellen.

Using the AWS Management Console

Um einen manuellen Snapshot mit dem zu erstellen AWS Management Console, können Sie eine der folgenden Methoden verwenden.

1. Methode 1:

    1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
    2. Wählen Sie im Navigationsbereich die Option Snapshots.

        > **Tip**
        >
        > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

    3. Wählen Sie auf der Seite Snapshots die Option Create (Erstellen) aus.
    4. Auf der Seite Create cluster snapshot (Cluster-Snapshot erstellen):
        a. Cluster-ID — Wählen Sie aus der Drop-down-Liste der Cluster den Cluster aus, von dem Sie einen Snapshot erstellen möchten.
        b. Snapshot-ID — Geben Sie einen Namen für Ihren Snapshot ein.

            Einschränkungen bei der Snapshot-Benennung:

            - Die Länge beträgt [1—255] Buchstaben, Zahlen oder Bindestriche.
            - Muss mit einem Buchstaben beginnen.

- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- Muss für alle Cluster (in Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

c. Wählen Sie Erstellen aus.

2. Methode 2:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie auf der Seite Cluster die Schaltfläche links neben dem Cluster aus, den Sie als Snapshot aufnehmen möchten.
4. Wählen Sie im Menü Actions (Aktionen) die Option Take snapshot (Snapshot erstellen) aus.
5. Auf der Seite Create cluster snapshot (Cluster-Snapshot erstellen):

a. Snapshot-ID — Geben Sie einen Namen für Ihren Snapshot ein.

Einschränkungen bei der Snapshot-Benennung:

- Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
- Muss mit einem Buchstaben beginnen.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- Muss für alle Cluster (in Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

b. Wählen Sie Erstellen aus.

Using the AWS CLI

Um einen Cluster-Snapshot mit dem zu erstellen AWS CLI, verwenden Sie den `create-db-cluster-snapshot` Vorgang mit den folgenden Parametern.

Parameter

- **`--db-cluster-identifier`** – Erforderlich. Der Name des Clusters, von dem Sie einen Snapshot machen. Dieser Cluster muss vorhanden sein und verfügbar sein.
- **`--db-cluster-snapshot-identifier`** – Erforderlich. Der Name des manuellen Snapshots, den Sie erstellen.

Im folgenden Beispiel wird ein Snapshot mit dem Namen `sample-cluster-snapshot` für einen Clusters namens `sample-cluster` erstellt.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Für Windows:

```
aws docdb create-db-cluster-snapshot ^
    --db-cluster-identifier sample-cluster ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-04-24T04:59:08.475Z",
```

```
        "Engine": "docdb",
        "Status": "creating",
        "Port": 0,
        "VpcId": "vpc-abc0123",
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:<accountID>:cluster-
snapshot:sample-cluster-snapshot"
    }
}
```

Amazon DocumentDB-Cluster-Snapshots kopieren

In Amazon DocumentDB können Sie manuelle und automatische Snapshots innerhalb desselben Kontos AWS-Region oder in ein anderes AWS-Region Konto kopieren. Sie können auch Schnappschüsse teilen, die anderen Personen AWS-Konten gehören. AWS-Region Es ist jedoch nicht möglich, einen Cluster-Snapshot AWS-Konto in AWS-Regionen einem einzigen Schritt zu kopieren. Diese Aktionen müssen einzeln ausgeführt werden.

Als Alternative zum Kopieren können Sie manuelle Schnappschüsse auch mit anderen AWS-Konten teilen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Snapshots teilen.

> **Note**
>
> Amazon DocumentDB berechnet Ihnen auf der Grundlage der Menge der Sicherungs- und Snapshot-Daten, die Sie aufbewahren, und des Zeitraums, in dem Sie sie aufbewahren. Weitere Informationen zum Speicher im Zusammenhang mit Amazon DocumentDB-Backups und -Snapshots finden Sie unter. Grundlegendes zur Sicherungsspeicher-Nutzung in Preisinformationen zum Amazon DocumentDB DocumentDB-Speicher finden Sie unter Amazon DocumentDB DocumentDB-Preise.

Themen

- Kopieren freigegebener Snapshots

- Schnappschüsse zwischen AWS-Regionen
- Einschränkungen
- Umgang mit Verschlüsselungen
- Überlegungen zu Parametergruppen
- Einen Cluster-Snapshot kopieren

Kopieren freigegebener Snapshots

Sie können Schnappschüsse kopieren, die Ihnen von anderen zur Verfügung gestellt wurden. AWS-Konten Wenn Sie einen verschlüsselten Snapshot kopieren, der von einem anderen geteilt wurde AWS-Konto, müssen Sie Zugriff auf den AWS KMS Verschlüsselungsschlüssel haben, mit dem der Snapshot verschlüsselt wurde.

Sie können einen gemeinsam genutzten Snapshot nur in denselben kopieren AWS-Region, unabhängig davon, ob der Snapshot verschlüsselt ist oder nicht. Weitere Informationen finden Sie unter Umgang mit Verschlüsselungen.

Schnappschüsse zwischen AWS-Regionen

Wenn Sie einen Snapshot in einen Snapshot kopieren AWS-Region , der sich von dem des Quell-Snapshots unterscheidet AWS-Region, ist jede Kopie ein vollständiger Snapshot. Eine vollständige Snapshot-Kopie enthält alle Daten und Metadaten, die für die Wiederherstellung des Amazon DocumentDB-Clusters erforderlich sind.

Je nach Umfang und AWS-Regionen Menge der zu kopierenden Daten kann es Stunden dauern, bis eine regionsübergreifende Snapshot-Kopie abgeschlossen ist. In einigen Fällen kann es zu einer großen Anzahl von regionsübergreifenden Snapshot-Kopieranfragen aus einer bestimmten Quelle kommen. AWS-Region In diesen Fällen kann Amazon DocumentDB neue regionsübergreifende Kopieranfragen von dieser Quelle AWS-Region in eine Warteschlange stellen, bis einige in Bearbeitung befindliche Kopien abgeschlossen sind. Zu Kopieranforderungen, die sich in der Warteschlange befinden, werden keine Fortschrittsinformationen angezeigt. Fortschrittsinformationen werden angezeigt, sobald der Kopiervorgang gestartet wird.

Einschränkungen

Im folgenden werden einige Einschränkungen beim Kopieren von Snapshots aufgeführt:

- Wenn Sie einen Quell-Snapshot löschen, bevor der Ziel-Snapshot verfügbar ist, kann das Kopieren des Snapshots fehlschlagen. Verifizieren Sie, dass der Ziel-Snapshot den Status `AVAILABLE` hat, bevor Sie einen Quell-Snapshot löschen.
- Pro Konto können bis zu fünf Snapshot-Kopieranforderungen an eine einzelne Zielregion aktiv sein.
- In Abhängigkeit von den beteiligten Regionen und der Menge der zu kopierenden Daten kann es Stunden dauern, bis eine regionsübergreifende Snapshot-Kopie fertiggestellt wird. Weitere Informationen finden Sie unter Schnappschüsse zwischen AWS-Regionen.

Umgang mit Verschlüsselungen

Sie können einen Snapshot kopieren, der mit einem AWS KMS -Verschlüsselungsschlüssel verschlüsselt wurde. Wenn Sie einen verschlüsselten Snapshot kopieren, muss auch die Kopie des Snapshots verschlüsselt werden. Wenn Sie einen verschlüsselten Snapshot innerhalb desselben Snapshots kopieren AWS-Region, können Sie die Kopie mit demselben AWS KMS Verschlüsselungsschlüssel wie den ursprünglichen Snapshot verschlüsseln, oder Sie können einen anderen Verschlüsselungsschlüssel angeben. AWS KMS Wenn Sie einen verschlüsselten Snapshot regionsübergreifend kopieren, können Sie für die Kopie nicht denselben AWS KMS Verschlüsselungsschlüssel verwenden wie für den Quell-Snapshot, da die AWS KMS Schlüssel regionsspezifisch sind. Stattdessen müssen Sie einen AWS KMS Schlüssel angeben, der für das Ziel AWS-Region n gültig ist.

Der Quell-Snapshot bleibt den gesamten Kopiervorgang über verschlüsselt. Weitere Informationen finden Sie unter Datenschutz in Amazon DocumentDB.

> **Note**
>
> Bei Amazon DocumentDB-Cluster-Snapshots können Sie einen unverschlüsselten Cluster-Snapshot nicht verschlüsseln, wenn Sie den Snapshot kopieren.

Überlegungen zu Parametergruppen

Wenn Sie einen Snapshot regionsübergreifend kopieren, enthält die Kopie nicht die Parametergruppe, die vom ursprünglichen Amazon DocumentDB-Cluster verwendet wurde. Wenn Sie einen Snapshot wiederherstellen, um einen neuen Cluster zu erstellen, erhält dieser Cluster die Standardparametergruppe für den Cluster, in dem AWS-Region er erstellt wurde. Um dem neuen Cluster dieselben Parameter wie dem Original zuzuweisen, müssen Sie wie folgt vorgehen:

1. Erstellen Sie im Ziel AWS-Region eine Amazon DocumentDB-Cluster-Parametergruppe mit denselben Einstellungen wie der ursprüngliche Cluster. Wenn im neuen System bereits eine vorhanden ist AWS-Region, können Sie diese verwenden.
2. Nachdem Sie den Snapshot im Ziel wiederhergestellt haben AWS-Region, ändern Sie den neuen Amazon DocumentDB-Cluster und fügen Sie die neue oder bestehende Parametergruppe aus dem vorherigen Schritt hinzu. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Einen Cluster-Snapshot kopieren

Sie können einen Amazon DocumentDB-Cluster mit dem AWS Management Console oder dem AWS CLI wie folgt kopieren.

Using the AWS Management Console

Gehen Sie wie folgt vor, um mit dem AWS Management Console eine Kopie eines Cluster-Snapshots zu erstellen. Dieses Verfahren eignet sich für das Kopieren von verschlüsselten oder unverschlüsselten Cluster-Snapshots in derselben Region AWS-Region oder in mehreren Regionen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Snapshots und dann die Schaltfläche links neben dem Snapshot, den Sie kopieren möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Actions die Option Copy aus.
4. Füllen Sie auf der daraufhin angezeigten Seite „Kopie des Cluster-Snapshots erstellen“ den Abschnitt Einstellungen aus.

   a. Zielregion — Optional. Um den Cluster-Snapshot in eine andere zu kopieren AWS-Region, wählen Sie diese Option AWS-Region für Zielregion aus.

b. Neue Snapshot-ID — Geben Sie einen Namen für den neuen Snapshot ein.

Einschränkungen bei der Benennung von Ziel-Snapshots:

- Kann nicht der Namen eines vorhandenen Snapshots sein.
- Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
- Muss mit einem Buchstaben beginnen.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.

c. Tags kopieren — Um alle Tags, die Sie in Ihrem Quell-Snapshot haben, in Ihre Snapshot-Kopie zu kopieren, wählen Sie Tags kopieren.

5. Füllen Sie den ncryption-at-rest Abschnitt E aus.

a. Verschlüsselung im Ruhezustand — Wenn Ihr Snapshot nicht verschlüsselt ist, stehen Ihnen diese Optionen nicht zur Verfügung, da Sie aus einem unverschlüsselten Snapshot keine verschlüsselte Kopie erstellen können. Wenn Ihr Snapshot verschlüsselt ist, können Sie die während der Verschlüsselung im Ruhezustand AWS KMS key verwendete Einstellung ändern.

Weitere Hinweise zur Verschlüsselung von Snapshot-Kopien finden Sie unterKopieren Sie die Cluster-Snapshot-.

Weitere Informationen zur Verschlüsselung im Ruhezustand finden Sie unter Verschlüsselung von Amazon DocumentDB DocumentDB-Daten im Ruhezustand.

b. AWS KMS Schlüssel — Wählen Sie aus der Dropdownliste eine der folgenden Optionen aus:

- (Standard) aws/rds — Die Kontonummer und die AWS KMS Schlüssel-ID werden nach dieser Option aufgeführt.
- < some-key-name > — Wenn Sie einen Schlüssel erstellt haben, wird er aufgelistet und steht Ihnen zur Auswahl zur Verfügung.
- Geben Sie einen Schlüssel-ARN ein — Geben Sie im Feld ARN den Amazon-Ressourcennamen (ARN) für Ihren AWS KMS Schlüssel ein. Das ARN-Format lautet `arn:aws:kms:<region>:<accountID>:key/<key-id>` .

6. Wenn Sie eine Kopie des ausgewählten Snapshots erstellen möchten, wählen Sie Copy Snapshot (Snapshot kopieren). Alternativ können Sie Abbrechen wählen, um keine Kopie des Snapshots zu erstellen.

Using the AWS CLI

Um mit dem eine Kopie eines unverschlüsselten Cluster-Snapshots zu erstellen AWS CLI, verwenden Sie den `copy-db-cluster-snapshot` Vorgang mit den folgenden Parametern. Wenn Sie den Snapshot in einen anderen kopieren AWS-Region, führen Sie den Befehl aus, in AWS-Region den der Snapshot kopiert werden soll.

- **`--source-db-cluster-snapshot-identifier`** – Erforderlich. Die ID des zu kopierenden Cluster-Snapshots, der kopiert werden soll. Der Cluster-Snapshot muss vorhanden sein und muss sich im verfügbaren Zustand befindet. Wenn Sie den Snapshot in einen anderen kopieren AWS-Region, muss dieser Bezeichner das ARN-Format für die Quelle haben AWS-Region. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- **`--target-db-cluster-snapshot-identifier`** – Erforderlich. Die ID des neuen Cluster-Snapshots, der aus dem Quell-Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

  Einschränkungen bei der Benennung von Ziel-Snapshots:

  - Kann nicht der Namen eines vorhandenen Snapshots sein.
  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.
- **`--source-region`**— Wenn Sie den Snapshot in einen anderen kopieren AWS-Region, geben Sie an, von welchem der AWS-Region verschlüsselte Cluster-Snapshot kopiert werden soll.

  Wenn Sie den Snapshot in einen anderen kopieren AWS-Region und nichts angeben`--source-region`, müssen Sie stattdessen die `pre-signed-url` Option angeben. Der `pre-signed-url` Wert muss eine URL sein, die eine mit Signature Version 4 signierte Anforderung enthält, damit die `CopyDBClusterSnapshot` Aktion in der Quelle AWS-Region aufgerufen

wird, aus der der Cluster-Snapshot kopiert wurde. Weitere Informationen zu finden Sie `pre-signed-url` unter DBClusterSnapshot kopieren.

- **`--kms-key-id`**— Die KMS-Schlüssel-ID für den Schlüssel, der zum Verschlüsseln der Kopie des Cluster-Snapshots verwendet werden soll.

  Wenn Sie einen verschlüsselten Cluster-Snapshot in einen anderen kopieren AWS-Region, ist dieser Parameter erforderlich. Sie müssen einen KMS-Schlüssel für das Ziel angeben AWS-Region.

  Wenn Sie einen verschlüsselten Cluster-Snapshot in dasselbe kopieren AWS-Region, ist der AWS KMS Schlüsselparameter optional. Die Kopie des Cluster-Snapshots wird mit demselben AWS KMS Schlüssel verschlüsselt wie der Quell-Cluster-Snapshot. Wenn Sie einen neuen AWS KMS Verschlüsselungsschlüssel angeben möchten, der zum Verschlüsseln der Kopie verwendet werden soll, können Sie dies mit diesem Parameter tun.
- **`--copy-tags`**— Fakultativ. Die Tags und Werte, die kopiert werden sollen.

Um einen laufenden Kopiervorgang abzubrechen, können Sie den Ziel-Cluster-Snapshot löschen, der dadurch identifiziert wurde, `--target-db-cluster-snapshot-identifier` oder `TargetDBClusterSnapshotIdentifier` während sich der Cluster-Snapshot im Kopierstatus befindet.

Example

Beispiel 1: Kopieren Sie einen unverschlüsselten Snapshot in dieselbe Region

Im folgenden AWS CLI Beispiel wird eine Kopie von `sample-cluster-snapshot` named `sample-cluster-snapshot-copy` in derselben Datei AWS-Region wie der Quell-Snapshot erstellt. Beim Erstellen der Kopie werden alle Tags des ursprünglichen Snapshots in die Snapshot-Kopie übernommen.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --copy-tags
```

Für Windows:

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
    --copy-tags
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-03-27T08:40:24.805Z",
        "Engine": "docdb",
        "Status": "copying",
        "Port": 0,
        "VpcId": "vpc-abcd0123",
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot"
    }
}
```

Example

Beispiel 2: Kopieren Sie einen unverschlüsselten Snapshot AWS-Regionen

Das folgende AWS CLI Beispiel erstellt eine Kopie von`sample-cluster-snapshot`, die den ARN hat`arn:aws:rds:us-east-1:123456789012:cluster-snapshot:sample-`

`cluster-snapshot`. Diese Kopie trägt den Namen named `sample-cluster-snapshot-copy` und befindet sich in der AWS-Region Datei, in der der Befehl ausgeführt wird.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy
```

Für Windows:

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "sample-cluster",
        "SnapshotCreateTime": "2020-04-29T16:45:51.239Z",
        "Engine": "docdb",
        "AllocatedStorage": 0,
        "Status": "copying",
        "Port": 0,
        "VpcId": "vpc-abc0123",
        "ClusterCreateTime": "2020-04-28T16:43:00.294Z",
        "MasterUsername": "master-user",
        "EngineVersion": "4.0.0",
        "LicenseModel": "docdb",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": false,
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
```

```
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot",
    }
}
```

Example

Beispiel 3: Kopieren Sie einen verschlüsselten Snapshot nach AWS-Regionen

Im folgenden AWS CLI Beispiel wird eine Kopie `sample-cluster-snapshot` von von der Region us-west-2 in die Region us-east-1 erstellt. Dieser Befehl wird in der Region us-east-1 aufgerufen.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot \
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy \
    --source-region us-west-2 \
    --kms-key-id sample-us-east-1-key
```

Für Windows:

```
aws docdb copy-db-cluster-snapshot ^
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
west-2:123456789012:cluster-snapshot:sample-cluster-snapshot ^
    --target-db-cluster-snapshot-identifier sample-cluster-snapshot-copy ^
    --source-region us-west-2 ^
    --kms-key-id sample-us-east-1-key
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshot": {
        "AvailabilityZones": [],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot-copy",
        "DBClusterIdentifier": "ayhu-xrsc-test-ap-southeast-1-small-cluster-kms",
        "SnapshotCreateTime": "2020-04-29T16:45:53.159Z",
        "Engine": "docdb",
        "AllocatedStorage": 0,
        "Status": "copying",
```

```
        "Port": 0,
        "ClusterCreateTime": "2020-04-28T16:43:07.129Z",
        "MasterUsername": "chimera",
        "EngineVersion": "4.0.0",
        "LicenseModel": "docdb",
        "SnapshotType": "manual",
        "PercentProgress": 0,
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/sample-key-id",
        "DBClusterSnapshotArn": "arn:aws:rds:us-east-1:111122223333:cluster-
snapshot:sample-cluster-snapshot-copy",
        "SourceDBClusterSnapshotArn": "arn:aws:rds:us-west-2:111122223333:cluster-
snapshot:sample-cluster-snapshot",
    }
}
```

> **Note**
>
> Weitere Hinweise zur Verschlüsselung von Snapshot-Kopien finden Sie unter. Kopieren Sie die Cluster-Snapshot-
> Weitere Informationen zur Verschlüsselung im Ruhezustand finden Sie unter Verschlüsselung von Amazon DocumentDB DocumentDB-Daten im Ruhezustand.

Amazon DocumentDB-Cluster-Snapshots teilen

Mit Amazon DocumentDB können Sie einen manuellen Cluster-Snapshot auf folgende Weise teilen:

- Durch die gemeinsame Nutzung eines manuellen Cluster-Snapshots, ob verschlüsselt oder unverschlüsselt, können autorisierte AWS Konten den Snapshot kopieren.
- Durch die gemeinsame Nutzung eines manuellen Cluster-Snapshots, ob verschlüsselt oder unverschlüsselt, können autorisierte AWS Konten einen Cluster direkt aus dem Snapshot wiederherstellen, anstatt eine Kopie davon zu erstellen und die Wiederherstellung von dort aus durchzuführen.

> **Note**
>
> Um einen automatisierten Cluster-Snapshot gemeinsam zu nutzen, erstellen Sie einen manuellen Cluster-Snapshot, indem Sie den automatisierten Snapshot kopieren, und geben

Sie diese Kopie dann weiter. Dieser Prozess gilt auch für durch AWS Backup generierte Ressourcen.

Sie können einen manuellen Snapshot mit bis zu 20 anderen teilen. AWS-Konten Sie können darüber hinaus einen nicht verschlüsselten Snapshot als öffentlich freigeben. Damit ist der Snapshot für alle -Konten verfügbar. Vergewissern Sie sich bei der Freigabe eines Snapshots als öffentlich darauf, dass in Ihren öffentlichen Snapshots keine privaten Informationen enthalten sind.

Wenn Sie manuelle Snapshots mit anderen AWS-Konten teilen und einen Cluster aus einem gemeinsam genutzten Snapshot mithilfe der AWS CLI oder der Amazon DocumentDB DocumentDB-API wiederherstellen, müssen Sie den Amazon-Ressourcennamen (ARN) des gemeinsam genutzten Snapshots als Snapshot-ID angeben.

Freigeben eines verschlüsselten Snapshots

Die folgenden Einschränkungen gelten für die Freigabe verschlüsselter Snapshots:

- Sie können verschlüsselte Snapshots nicht als öffentlich freigeben.
- Sie können keinen Snapshot teilen, der mit dem AWS KMS Standardverschlüsselungsschlüssel des Kontos verschlüsselt wurde, das den Snapshot geteilt hat.

Führen Sie die folgenden Schritte aus, um verschlüsselte Snapshots freizugeben.

1. Teilen Sie den Verschlüsselungsschlüssel AWS Key Management Service (AWS KMS), der zum Verschlüsseln des Snapshots verwendet wurde, mit allen Konten, die auf den Snapshot zugreifen möchten.

   Sie können AWS KMS Verschlüsselungsschlüssel mit anderen AWS Konten teilen, indem Sie die anderen Konten zur AWS KMS Schlüsselrichtlinie hinzufügen. Einzelheiten zur Aktualisierung einer Schlüsselrichtlinie finden Sie unter Verwenden von Schlüsselrichtlinien in AWS KMS im AWS Key Management Service Entwicklerhandbuch. Ein Beispiel für die Erstellung einer Schlüsselrichtlinie finden Sie unter Erstellen einer IAM-Richtlinie, um das Kopieren des verschlüsselten Snapshots zu erlauben an späterer Stelle in diesem Thema.
2. Verwenden Sie AWS CLI, wie unten gezeigt, um den verschlüsselten Snapshot mit den anderen Konten zu teilen.

Zugriff auf einen AWS KMS Verschlüsselungsschlüssel zulassen

Damit ein anderer AWS-Konto einen verschlüsselten Snapshot, der von Ihrem Konto geteilt wurde, kopieren kann, muss das Konto, mit dem Sie Ihren Snapshot teilen, Zugriff auf den AWS KMS Schlüssel haben, mit dem der Snapshot verschlüsselt wurde. Um einem anderen Konto Zugriff auf einen AWS KMS Schlüssel zu gewähren, aktualisieren Sie die Schlüsselrichtlinie für den AWS KMS Schlüssel mit dem ARN des Kontos, das Sie als Principal teilen, in der AWS KMS Schlüsselrichtlinie. Lassen Sie anschließend die Aktion `kms:CreateGrant` zu.

Nachdem Sie einem Konto Zugriff auf Ihren AWS KMS Verschlüsselungsschlüssel gewährt haben, muss dieses Konto, um Ihren verschlüsselten Snapshot zu kopieren, einen AWS Identity and Access Management (IAM-) Benutzer erstellen, falls es noch keinen hat. Darüber hinaus muss dieses Konto diesem IAM-Benutzer eine IAM-Richtlinie zuordnen, die es dem Benutzer ermöglicht, mithilfe Ihres Schlüssels einen verschlüsselten Snapshot zu kopieren. AWS KMS Das Konto muss ein IAM-Benutzer sein und darf aufgrund von Sicherheitseinschränkungen keine AWS-Konto Root-Identität sein. AWS KMS

Im folgenden Beispiel für eine Schlüsselrichtlinie ist der Benutzer 123451234512 der Besitzer des Verschlüsselungsschlüssels. AWS KMS Bei Benutzer 123456789012 handelt es sich um das Konto, für das der Schlüssel freigegeben wird. Diese aktualisierte Schlüsselrichtlinie gewährt dem Konto Zugriff auf den Schlüssel. AWS KMS Dazu wird der ARN für die AWS-Konto Root-Identität für den Benutzer 123456789012 als Principal für die Richtlinie aufgenommen und die Aktion zugelassen. `kms:CreateGrant`

```
{
    "Id": "key-policy-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::123451234512:user/KeyUser",
                "arn:aws:iam::123456789012:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
```

```
                "kms:DescribeKey"
            ],
            "Resource": "*"},
            {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {"AWS": [
                "arn:aws:iam::123451234512:user/KeyUser",
                "arn:aws:iam::123456789012:root"
            ]},
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
        }
    ]
}
```

Erstellen einer IAM-Richtlinie, um das Kopieren des verschlüsselten Snapshots zu erlauben

Wenn der externe Benutzer Zugriff auf Ihren AWS KMS Schlüssel AWS-Konto hat, kann der Besitzer dieses Kontos eine Richtlinie erstellen, die es einem IAM-Benutzer, der für das Konto erstellt wurde, ermöglicht, einen verschlüsselten Snapshot zu kopieren, der mit diesem Schlüssel verschlüsselt ist. AWS KMS

Das folgende Beispiel zeigt eine Richtlinie, die einem IAM-Benutzer für AWS-Konto 123456789012 zugewiesen werden kann. Die Richtlinie ermöglicht es dem IAM-Benutzer, einen gemeinsam genutzten Snapshot aus dem Konto 123451234512 zu kopieren, der mit dem AWS KMS Schlüssel `c989c1dd-a3f2-4a5d-8d96-e793d082ab26` in der Region us-west-2 verschlüsselt wurde.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUseOfTheKey",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
```

```
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey",
                "kms:CreateGrant",
                "kms:RetireGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"]
        },
        {
            "Sid": "AllowAttachmentOfPersistentResources",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": ["arn:aws:kms:us-west-2:123451234512:key/c989c1dd-
a3f2-4a5d-8d96-e793d082ab26"],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

Einzelheiten zur Aktualisierung einer Schlüsselrichtlinie finden Sie unter Verwenden von Schlüsselrichtlinien im Entwicklerhandbuch. AWS KMSAWS Key Management Service

Freigeben eines Snapshots

Sie können einen manuellen Amazon DocumentDB-Cluster-Snapshot (oder eine Kopie eines automatisierten Snapshots) mit dem AWS Management Console oder dem AWS CLI teilen:

Using the AWS Management Console

Gehen Sie wie folgt vor AWS Management Console, um einen Snapshot mit dem zu teilen:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Wählen Sie im Navigationsbereich die Option Snapshots.
3. Wählen Sie den manuellen Snapshot, den Sie freigeben möchten.
4. Wählen Sie im Drop-down-Menü Aktionen die Option Teilen aus.
5. Wählen Sie eine der folgenden Optionen für die Sichtbarkeit von DB-Snapshots:

   - Wenn die Quelle unverschlüsselt ist, wählen Sie Öffentlich, damit alle AWS Konten einen Cluster aus Ihrem manuellen Snapshot wiederherstellen können. Oder wählen Sie Privat, um nur AWS Konten zuzulassen, die Sie für die Wiederherstellung eines Clusters aus Ihrem manuellen Snapshot angeben.

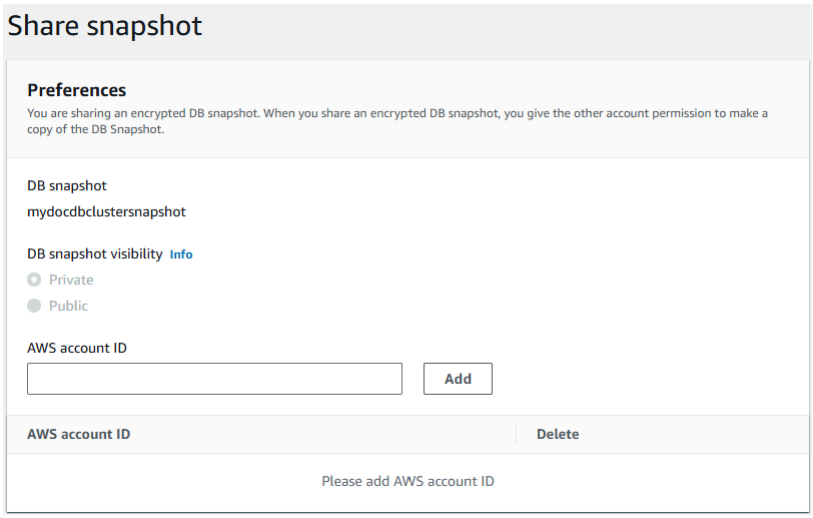
     > **Warning**
     >
     > Wenn Sie die Sichtbarkeit von DB-Snapshots auf Öffentlich setzen, können alle AWS Konten einen Cluster aus Ihrem manuellen Snapshot wiederherstellen und haben Zugriff auf Ihre Daten. Geben Sie keine manuellen Cluster-Snapshots, die private Informationen enthalten, als öffentlich frei.

   - Ist die Quelle verschlüsselt, ist DB Snapshot Visibility (Sichtbarkeit des DB-Snapshots) auf Private (Privat) festgelegt, da verschlüsselte Snapshots nicht als öffentlich freigegeben werden können.

     > **Note**
     >
     > Snapshots, die mit der Standardeinstellung verschlüsselt wurden, AWS KMS key können nicht geteilt werden.

6. Geben Sie AWS unter Konto-ID die AWS Konto-ID für ein Konto ein, dem Sie die Wiederherstellung eines Clusters aus Ihrem manuellen Snapshot erlauben möchten, und wählen Sie dann Hinzufügen aus. Wiederholen Sie den Vorgang, um weitere AWS Kontokennungen hinzuzufügen, bis zu 20 AWS Konten.

   Wenn Ihnen beim Hinzufügen einer AWS Konto-ID zur Liste der zulässigen Konten ein Fehler unterläuft, können Sie diese aus der Liste löschen, indem Sie rechts neben der falschen AWS Konto-ID die Option Löschen auswählen.



7. Nachdem Sie Kennungen für alle AWS Konten hinzugefügt haben, denen Sie die Wiederherstellung des manuellen Snapshots erlauben möchten, wählen Sie Speichern, um Ihre Änderungen zu speichern.

Using the AWS CLI

Um einen Snapshot mit dem zu teilen AWS CLI, verwenden Sie den Amazon DocumentDB `modify-db-snapshot-attribute` DocumentDB-Vorgang. Verwenden Sie den `--values-to-add` Parameter, um eine Liste der Personen IDs hinzuzufügen AWS-Konten , die berechtigt sind, den manuellen Snapshot wiederherzustellen.

Im folgenden Beispiel sind zwei AWS-Konto Kennungen, 123451234512 und 123456789012, für die Wiederherstellung des genannten Snapshots zulässig. `manual-snapshot1` Außerdem wird der `all`-Attributwert entfernt, um den Snapshot als privat zu markieren.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add '["123451234512","123456789012"]'
```

Für Windows:

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
```

```
    --values-to-add '["123451234512","123456789012"]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512",
                    "123456789012"
                ]
            }
        ]
    }
}
```

Verwenden Sie den Parameter, um einen AWS-Konto Bezeichner aus der Liste zu entfernen. `--values-to-remove` Das folgende Beispiel verhindert, dass AWS-Konto ID 123456789012 den Snapshot wiederherstellt.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove '["123456789012"]'
```

Für Windows:

```
aws docdb modify-db-cluster-snapshot-attribute ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot ^
    --attribute-name restore ^
    --values-to-remove '["123456789012"]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot",
```

```
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123451234512"
                ]
            }
        ]
    }
}
```

Wiederherstellung aus einem Cluster-Snapshot

Amazon DocumentDB (mit MongoDB-Kompatibilität) erstellt einen Cluster-Snapshot Ihres Speichervolumes. Sie können einen neuen Cluster erstellen, indem Sie ihn aus einem Cluster-Snapshot wiederherstellen. Wenn Sie den Cluster wiederherstellen, geben Sie den Namen des wiederherzustellenden Cluster-Snapshots und einen Namen für den neuen Cluster an, der durch die Wiederherstellung erstellt wird. Sie können nicht von einem Snapshot auf einen bestehenden Cluster wiederherstellen, da bei der Wiederherstellung ein neuer Cluster erstellt wird.

Wenn Sie einen Cluster aus einem Cluster-Snapshot wiederherstellen:

- Diese Aktion stellt nur den Cluster wieder her, nicht die Instances für diesen Cluster. Sie müssen die Aktion `create-db-instance` aufrufen, um Instances für den wiederhergestellten Cluster zu erstellen, wobei Sie in `--db-cluster-identifier` die ID des wiederhergestellten Clusters angeben. Sie können Instances erst erstellen, nachdem der Cluster den Status available hat.
- Sie können einen verschlüsselten Snapshot nicht in einem unverschlüsselten Cluster wiederherstellen. Sie können jedoch einen unverschlüsselten Snapshot in einem verschlüsselten Cluster wiederherstellen, indem Sie den Schlüssel angeben. AWS KMS
- Um einen Cluster aus einem verschlüsselten Snapshot wiederherzustellen, müssen Sie Zugriff auf den AWS KMS Schlüssel haben.

> **Note**
>
> Sie können einen 3.6-Cluster nicht auf einem 4.0-Cluster wiederherstellen, aber Sie können von einer Clusterversion zu einer anderen migrieren. Weitere Informationen finden Sie unter Migration zu Amazon DocumentDB.

Using the AWS Management Console

Das folgende Verfahren zeigt, wie Sie mit der Amazon DocumentDB Management Console einen Amazon DocumentDB-Cluster aus einem Cluster-Snapshot wiederherstellen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Snapshots und dann die Schaltfläche links neben dem Snapshot aus, mit der Sie einen Cluster wiederherstellen möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Aktionen die Option Restore (Wiederherstellen).
4. Füllen Sie auf der Seite Restore snapshot (Snapshot wiederherstellen) den Abschnitt Configuration (Konfiguration) aus.

   a. Cluster-ID — Der Name für den neuen Cluster. Sie können den von Amazon DocumentDB angegebenen Namen akzeptieren oder einen Namen eingeben, den Sie bevorzugen. Der DBsupplied Amazon-Dokumentname hat das Format `docdb-` plus einen UTC-Zeitstempel; zum Beispiel`docdb-`*`yyyy-mm-dd-hh-mm-ss`*.

   b. Instance-Klasse — Die Instance-Klasse für den neuen Cluster. Sie können die Standard-Instance-Klasse akzeptieren oder eine Instance-Klasse aus der Dropdown-Liste auswählen.

   c. Anzahl der Instanzen — Die Anzahl der Instanzen, die Sie mit diesem Cluster erstellen möchten. Sie können den Standard von 3 Instanzen (1 primäres read/write und 2 schreibgeschützte Replikate) akzeptieren oder die Anzahl der Instanzen aus der Drop-down-Liste wählen.

5. Wählen Sie für die Cluster-Speicherkonfiguration eine Speicheroption aus.

> **Note**
>
> Die I/O-optimierte Speicherkonfiguration von Amazon DocumentDB ist nur in der Engine-Version Amazon DocumentDB 5.0 verfügbar.

6. Wenn Sie mit der Cluster-Konfiguration zufrieden sind, wählen Sie Restore cluster (Cluster wiederherstellen) aus und warten Sie, bis Ihr Cluster wiederhergestellt ist.
7. Wenn Sie es vorziehen, einige Konfigurationen zu ändern, z. B. eine nicht standardmäßige Amazon VPC oder Sicherheitsgruppe anzugeben, wählen Sie unten links auf der Seite Erweiterte Einstellungen anzeigen aus und fahren Sie dann mit den folgenden Schritten fort.

   a. Vervollständigen Sie den Bereich Network settings (Netzwerkeinstellungen).

      - Virtual Private Cloud (VPC) — Akzeptieren Sie die aktuelle VPC oder wählen Sie eine VPC aus der Drop-down-Liste aus.
      - Subnetzgruppe — Akzeptieren Sie die `default` Subnetzgruppe oder wählen Sie eine aus der Drop-down-Liste aus.
      - VPC-Sicherheitsgruppen — Akzeptieren Sie die `default (VPC)` Sicherheitsgruppe oder wählen Sie eine aus der Liste aus.

   b. Vervollständigen Sie den Abschnitt Cluster options (Cluster-Optionen).

      - Datenbankport — Akzeptieren Sie den Standardport`27017`, oder verwenden Sie den Aufwärts- oder Abwärtspfeil, um den Port festzulegen, den Sie für Anwendungsverbindungen verwenden möchten.

   c. Vervollständigen Sie den Abschnitt Encryption (Verschlüsselung).

      - Verschlüsselung im Ruhezustand — Wenn Ihr Snapshot verschlüsselt ist, stehen Ihnen diese Optionen nicht zur Verfügung. Wenn er nicht verschlüsselt ist, können Sie eine der folgenden Optionen auswählen:
        - Um alle Daten Ihres Clusters zu verschlüsseln, wählen Sie Aktivieren encryption-at-rest. Wenn Sie diese Option wählen, müssen Sie einen KMS-Schlüssel angeben.
        - Um die Daten Ihres Clusters nicht zu verschlüsseln, wählen Sie Disable. encryption-at-rest Wenn Sie diese Option wählen, sind Sie mit dem Abschnitt "Verschlüsselung" fertig.
      - AWS KMS Schlüssel — Wählen Sie eine der folgenden Optionen aus der Drop-down-Liste aus:

- (Standard) aws/rds — Die Kontonummer und die AWS KMS Schlüssel-ID werden nach dieser Option aufgeführt.
- Vom Kunden verwalteter Schlüssel — Diese Option ist nur verfügbar, wenn Sie einen IAM-Verschlüsselungsschlüssel in der AWS Identity and Access Management (IAM-) Konsole erstellt haben. Sie können den Schlüssel für die Verschlüsselung Ihres Clusters auswählen.
- Geben Sie einen Schlüssel-ARN ein — Geben Sie im Feld ARN den Amazon-Ressourcennamen (ARN) für Ihren AWS KMS Schlüssel ein. Das ARN-Format lautet `arn:aws:kms:<region>:<accountID>:key/<key-id>`.

d. Füllen sie den Abschnitt Log exports (Protokollexporte) aus.

- Wählen Sie die Protokolltypen aus, in denen veröffentlicht werden soll CloudWatch — Wählen Sie eine der folgenden Optionen:
  - Aktiviert — Ermöglicht Ihrem Cluster den Export von DDL-Protokollen nach Amazon CloudWatch Logs.
  - Deaktiviert — Verhindert, dass Ihr Cluster DDL-Logs nach Amazon CloudWatch Logs exportiert. Disabled (Deaktiviert) ist die Voreinstellung.
- IAM-Rolle — Wählen Sie aus der Liste die Option RDS Service Linked Role aus.

e. Füllen Sie den Abschnitt Tags aus.

- Tag hinzufügen — Geben Sie im Feld Schlüssel den Namen für das Tag für Ihren Cluster ein. Geben Sie optional im Feld Value (Wert) den Tag-Wert ein. Tags werden zusammen mit AWS Identity and Access Management (IAM-) Richtlinien verwendet, um den Zugriff auf Amazon DocumentDB DocumentDB-Ressourcen zu verwalten und zu kontrollieren, welche Aktionen auf die Ressourcen angewendet werden können.

f. Füllen Sie den Abschnitt Deletion protection (Löschschutz) aus.

- Löschschutz aktivieren — Schützt den Cluster davor, versehentlich gelöscht zu werden. Wenn diese Option aktiviert ist, können Sie den Cluster nicht löschen.

8. Wählen Sie Restore cluster (Cluster wiederherstellen) aus.

Using the AWS CLI

Um einen Cluster mithilfe von aus einem Snapshot wiederherzustellen AWS CLI, verwenden Sie den `restore-db-cluster-from-snapshot` Vorgang mit den folgenden Parametern. Weitere Informationen finden Sie unter RestoreDBClusterFromSnapshot.

- **`--db-cluster-identifier`** – Erforderlich. Der Name des Clusters, der von der Operation erstellt wird. Ein Cluster mit diesem Namen darf vor dieser Operation nicht vorhanden sein.

  Einschränkungen bei der Benennung von Clustern:

  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.
- **`--snapshot-identifier`** – Erforderlich. Der Name des Snapshots, der für die Wiederherstellung verwendet wird. Ein Snapshot mit diesem Namen muss vorhanden sein und sich im verfügbaren Zustand befindet.
- **`--engine`** – Erforderlich. Der Wert muss `docdb` sein.
- **`--storage-type standard | iopt1`**— Fakultativ. Standard: `standard`.
- **`--kms-key-id`**— Fakultativ. Der ARN der AWS KMS Schlüssel-ID, die beim Wiederherstellen eines verschlüsselten Snapshots oder beim Verschlüsseln eines Clusters bei der Wiederherstellung aus einem unverschlüsselten Snapshot verwendet werden soll. Die Angabe der AWS KMS Schlüssel-ID führt dazu, dass der wiederhergestellte Cluster mit dem AWS KMS Schlüssel verschlüsselt wird, unabhängig davon, ob der Snapshot verschlüsselt wurde oder nicht.

  Das Format der `--kms-key-id` ist `arn:aws:kms:<region>:<accountID>:key/<key-id>`. Wenn Sie keinen Wert für den `--kms-key-id`-Parameter angeben, geschieht folgendes:

  - Wenn der Snapshot verschlüsselt `--snapshot-identifier` ist, wird der wiederhergestellte Cluster mit demselben AWS KMS Schlüssel verschlüsselt, der zum Verschlüsseln des Snapshots verwendet wurde.
  - Wenn der Snapshot in `--snapshot-identifier` nicht verschlüsselt ist, dann ist der wiederhergestellte Cluster nicht verschlüsselt.

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-from-snapshot \
    --db-cluster-identifier sample-cluster-restore \
    --snapshot-identifier sample-cluster-snapshot \
```

```
    --engine docdb \
    --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

Für Windows:

```
aws docdb restore-db-cluster-from-snapshot ^
    --db-cluster-identifier sample-cluster-restore ^
    --snapshot-identifier sample-cluster-snapshot ^
    --engine docdb ^
    --kms-key-id arn:aws:kms:us-east-1:123456789012:key/SAMPLE-KMS-KEY-ID
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster-restore",
        "DBClusterParameterGroup": "default.docdb4.0",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-restore.cluster-node.us-
east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "Port": 27017,
        "MasterUsername": "<master-user>",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcdefgh",
                "Status": "active"
            }
        ],
```

```
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-
restore",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-04-01T01:43:40.871Z",
        "DeletionProtection": true
    }
}
```

Nachdem der Clusterstatus available lautet, erstellen Sie mindestens eine Instance für den Cluster.

Für Linux, macOS oder Unix:

```
aws docdb create-db-instance \
    --db-cluster-identifier sample-cluster-restore  \
    --db-instance-identifier sample-cluster-restore-instance \
    --availability-zone us-east-1b \
    --promotion-tier 2 \
    --db-instance-class db.r5.large \
    --engine docdb
```

Für Windows:

```
aws docdb create-db-instance ^
    --db-cluster-identifier sample-cluster-restore  ^
    --db-instance-identifier sample-cluster-restore-instance ^
    --availability-zone us-east-1b ^
    --promotion-tier 2 ^
    --db-instance-class db.r5.large ^
    --engine docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-cluster-restore-instance",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
```

```
        "DBInstanceStatus": "creating",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcdefgh",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-west-2b",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcdefgh",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    ...
                }
            ]
        },
        "PreferredMaintenanceWindow": "fri:09:43-fri:10:13",
        "PendingModifiedValues": {},
        "EngineVersion": "4.0.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster-restore",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/<sample-key-id>",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 2,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-cluster-
restore-instance"
    }
}
```

Zu einem bestimmten Zeitpunkt wiederherstellen

Sie können einen Cluster zu einem beliebigen Zeitpunkt innerhalb der Aufbewahrungsfrist für Backups des Clusters wiederherstellen, indem Sie AWS Management Console oder AWS Command Line Interface (AWS CLI) verwenden.

> **Note**
>
> Sie können keine point-in-time Wiederherstellung eines 3.6-Clusters auf einen 4.0-Cluster durchführen, aber Sie können von einer Clusterversion zu einer anderen migrieren. Weitere Informationen finden Sie unter Migration zu Amazon DocumentDB.

Beachten Sie Folgendes, wenn Sie einen Cluster zu einem bestimmten Zeitpunkt wiederherstellen.

- Der neue Cluster wird mit der gleichen Konfiguration wie der Quell-Cluster erstellt, nur dass der neue Cluster mit der Standard-Parametergruppe erstellt wird. Um die Parametergruppe des neuen Clusters auf die Parametergruppe des Quellclusters festzulegen, ändern Sie den Cluster, nachdem er den Status available (verfügbar) hat. Weitere Informationen zum Ändern eines Clusters finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Using the AWS Management Console

Sie können einen Cluster point-in-time innerhalb seines Aufbewahrungszeitraums für Backups auf a wiederherstellen, indem Sie die folgenden Schritte mit dem ausführen AWS Management Console.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Cluster, den Sie wiederherstellen möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol

(≡)
aus.

3. Wählen Sie im Menü Actions (Aktionen) die Option Restore to point in time (Zu einem bestimmten Zeitpunkt wiederherstellen).
4. Geben Sie im Bereich Restore time (Wiederherstellungszeit) Datum und Uhrzeit für die zeitpunktbezogene Wiederherstellung an.

   a. Wiederherstellungsdatum — Wählen Sie ein Datum, das zwischen der frühesten Wiederherstellungszeit und der spätesten Wiederherstellungszeit liegt, oder geben Sie ein Datum ein.

   b. Wiederherstellungszeit — Wählen Sie die Stunde, Minute und Sekunden, die zwischen der frühesten Wiederherstellungszeit und der spätesten Wiederherstellungszeit liegen, oder geben Sie sie ein.
5. Füllen Sie den Bereich Configuration (Konfiguration) aus.

   a. Cluster-ID — Akzeptieren Sie die Standard-ID, oder geben Sie eine von Ihnen bevorzugte ID ein.

      Einschränkungen bei der Benennung von Clustern:

      - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.

   b. Instance-Klasse — Wählen Sie aus der Drop-down-Liste die Instance-Klasse aus, die Sie für die Instances des Clusters verwenden möchten.

   c. Anzahl der Instanzen — Wählen Sie aus der Drop-down-Liste die Anzahl der Instanzen aus, die bei der Wiederherstellung des Clusters erstellt werden sollen.
6. Wählen Sie für die Cluster-Speicherkonfiguration eine Speicheroption aus.

> **Note**
>
> Die I/O-optimierte Speicherkonfiguration von Amazon DocumentDB ist nur in der Engine-Version Amazon DocumentDB 5.0 verfügbar.

7. Optional. Um die Netzwerkeinstellungen, Cluster-Optionen und die Aktivierung von Protokollexporten zu konfigurieren, wählen Sie Show advanced settings (Erweiterte Einstellungen anzeigen) aus. Schließen Sie dann die folgenden Abschnitte ab. Fahren Sie andernfalls mit dem nächsten Schritt fort.

   - Network settings (Netzwerkeinstellungen)

     1. Virtual Private Cloud (VPC) — Wählen Sie aus der Drop-down-Liste die VPC aus, die Sie für diesen Cluster verwenden möchten.
     2. Subnetzgruppe — Wählen Sie aus der Drop-down-Liste die Subnetzgruppe für diesen Cluster aus.
     3. VPC-Sicherheitsgruppen — Wählen Sie aus der Dropdownliste die VPC-Sicherheitsgruppen für diesen Cluster aus.

   - Cluster options (Cluster-Optionen)

     1. Port — Akzeptieren Sie den Standardport (27017) oder verwenden Sie die Aufwärts- und Abwärtspfeile, um den Port für die Kommunikation mit diesem Cluster festzulegen.

   - Protokollexporte

     1. Audit-Logs — Wählen Sie diese Option, um den Export von Audit-Logs nach Amazon CloudWatch Logs zu ermöglichen. Wenn Sie diese Option auswählen, müssen Sie `audit_logs` in der benutzerdefinierten Parametergruppe des Clusters aktivieren. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.
     2. Profiler-Protokolle — Wählen Sie diese Option, um den Export von Operation-Profiler-Protokollen nach Amazon CloudWatch Logs zu ermöglichen. Wenn Sie diese Option auswählen, müssen Sie auch die folgenden Parameter in der benutzerdefinierten Parametergruppe des Clusters ändern:
        - `profiler`— Eingestellt auf. `enabled`

- `profiler_threshold_ms`— Auf einen Wert setzen`[0-INT_MAX]`, um den Schwellenwert für Profilierungsvorgänge festzulegen.
- `profiler_sampling_rate`— Geben Sie einen Wert ein, `[0.0-1.0]` um den Prozentsatz langsamer Operationen festzulegen, für den ein Profil erstellt werden soll.

Weitere Informationen finden Sie unter Profilierung von Amazon DocumentDB DocumentDB-Vorgängen.

3. Profiler-Protokolle — Exportieren Sie Profiler-Protokolle nach Amazon CloudWatch
4. IAM-Rolle — Wählen Sie aus der Drop-down-Liste die Option RDS Service Linked Role aus.

- Tags
  1. Tag hinzufügen — Geben Sie im Feld Schlüssel den Namen für das Tag für Ihren Cluster ein. Geben Sie optional im Feld Value (Wert) den Tag-Wert ein. Tags werden zusammen mit AWS Identity and Access Management (IAM-) Richtlinien verwendet, um den Zugriff auf Amazon DocumentDB DocumentDB-Ressourcen zu verwalten und zu kontrollieren, welche Aktionen auf die Ressourcen angewendet werden können.
- Löschschutz
  1. Löschschutz aktivieren — Schützt den Cluster davor, versehentlich gelöscht zu werden. Wenn diese Option aktiviert ist, können Sie den Cluster nicht löschen.

8. Um den Cluster wiederherzustellen, wählen Sie Create cluster (Cluster erstellen). Alternativ können Sie Cancel (Abbrechen) auswählen, um den Vorgang abzubrechen.

Using the AWS CLI

Um einen Cluster zu einem bestimmten Zeitpunkt mit Hilfe der Aufbewahrungsdauer des Snapshots wiederherzustellen, verwenden Sie die Operation `restore-db-cluster-to-point-in-time` mit den folgenden Parametern.

- **`--db-cluster-identifier`**— Erforderlich. Der Name des neuen Clusters, der erstellt werden soll. Dieser Cluster darf vor der Operation nicht vorhanden sein. Der Parameterwert muss den folgenden Einschränkungen entsprechen.

  Einschränkungen bei der Benennung von Clustern:

  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.

  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.
- **`--restore-to-time`**— Das UTC-Datum und die Uhrzeit für die Wiederherstellung des Clusters. Beispiel, `2018-06-07T23:45:00Z`.

  Zeiteinschränkungen:
  - Muss vor dem letzten wiederherstellbaren Zeitpunkt für den Cluster liegen.
  - Muss angegeben werden, wenn der Parameter `--use-latest-restorable-time` nicht angegeben ist.
  - Kann nicht angegeben werden, wenn der Parameter `--use-latest-restorable-time` auf `true` festgelegt ist.
  - Kann nicht angegeben werden, wenn der Parameterwert `--restore-type` `copy-on-write` ist.
- **`--source-db-cluster-identifier`**— Der Name des Quell-Clusters, aus dem wiederhergestellt werden soll. Dieser Cluster muss vorhanden und verfügbar sein.
- **`--use-latest-restorable-time`**oder **`--no-use-latest-restorable-time`** — Ob die Wiederherstellung auf die letzte wiederherstellbare Backup-Zeit erfolgen soll. Darf nicht angegeben werden, wenn der Parameter `--restore-to-time` angegeben ist.
- **`--storage-type standard | iopt1`**— Fakultativ. Standard: `standard`.

Der AWS CLI Vorgang stellt `restore-db-cluster-to-point-in-time` nur den Cluster wieder her, nicht die Instanzen für diesen Cluster. Sie müssen die Operation `create-db-instance` aufrufen, um Instances für den wiederhergestellten Cluster zu erstellen, wobei Sie in `--db-cluster-identifier` die ID des wiederhergestellten Clusters angeben. Sie können Instances erst erstellen, nachdem die Operation `restore-db-cluster-to-point-in-time` abgeschlossen wurde und wenn der wiederhergestellte Cluster verfügbar ist.

Example

Das folgende Beispiel erstellt `sample-cluster-restored` aus Snapshot `sample-cluster-snapshot` für den letzten wiederherstellbaren Zeitpunkt.

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-restored \
    --source-db-cluster-identifier sample-cluster-snapshot \
    --use-latest-restorable-time
```

Für Windows:

```
aws docdb restore-db-cluster-to-point-in-time ^
    --db-cluster-identifier sample-cluster-restored ^
    --source-db-cluster-identifier sample-cluster-snapshot ^
    --use-latest-restorable-time
```

Example

Das folgende Beispiel erstellt `sample-cluster-restored` aus Snapshot `sample-cluster-snapshot` für 03:15 Uhr am 11. Dezember 2018 (UTC) (liegt innerhalb des Aufbewahrungszeitraums für Backups von `sample-cluster`).

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-to-point-in-time \
    --db-cluster-identifier sample-cluster-restore \
    --source-db-cluster-identifier sample-cluster \
    --restore-to-time 2020-05-12T03:15:00Z
```

Für Windows:

```
aws docdb restore-db-cluster-to-point-in-time ^
    --db-cluster-identifier sample-cluster-restore ^
    --source-db-cluster-identifier sample-cluster ^
    --restore-to-time 2020-05-12T03:15:00Z
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-west-2b",
            "us-west-2a"
```

```
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster-restored",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-restored.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-restored.node.us-
east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
        "Port": 27017,
        "MasterUsername": "master-user",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abc0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID^>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster-
restored",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-04-24T20:14:36.713Z",
        "DeletionProtection": false
    }
}
```

Löschen eines Cluster-Snapshots

Ein manueller Snapshot ist ein vollständiges Backup, das nur gelöscht wird, wenn Sie es manuell mit dem AWS Management Console oder AWS CLI löschen. Sie können einen automatischen Snapshot nicht manuell löschen, da automatische Snapshots nur gelöscht werden, wenn die Aufbewahrungsfrist des Snapshots abläuft oder Sie den Cluster des Snapshots löschen.

Using the AWS Management Console

Gehen Sie wie folgt vor AWS Management Console, um einen manuellen Cluster-Snapshot mithilfe von zu löschen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie in der Liste der Snapshots die Schaltfläche links neben dem Snapshot, den Sie löschen möchten. Der Typ des Snapshots muss manuell sein.

   1. Sie können überprüfen, ob der Snapshot-Typ manuell ist, indem Sie überprüfen, ob er als **`manual`** oder in der **`automatic`** Spalte Typ aufgeführt ist.
4. Wählen Sie im Menü Actions (Aktionen) die Option Delete (Löschen) aus. Wenn die Option Delete (Löschen) nicht verfügbar ist, haben Sie wahrscheinlich einen automatischen Snapshot auswählt.
5. Um den Snapshot zu löschen, wählen Sie auf der Bestätigungsseite des Löschvorgangs Delete (Löschen) aus. Um den Snapshot zu erhalten, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Ein manueller Amazon DocumentDB-Cluster-Snapshot ist ein vollständiges Backup, das Sie manuell mit dem AWS CLI löschen können. Sie können einen automatischen Snapshot nicht manuell löschen.

Um einen manuellen Cluster-Snapshot mit dem zu löschen AWS CLI, verwenden Sie den `delete-db-cluster-snapshot` Vorgang mit den folgenden Parametern.

Parameter

- **`--db-cluster-snapshot-identifier`** – Erforderlich. Der Name des zu löschenden manuellen Snapshots.

Das folgende Beispiel löscht den Cluster-Snapshot `sample-cluster-snapshot`.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Für Windows:

```
aws docdb delete-db-cluster-snapshot ^
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Die Ausgabe dieses Vorgangs führt die Einzelheiten des von Ihnen gelöschten Cluster-Snapshots auf.

Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen

In diesen Abschnitten werden die verschiedenen Komponenten und die damit verbundenen Aufgaben für die Verwaltung Ihrer Amazon DocumentDB DocumentDB-Implementierung (mit MongoDB-Kompatibilität) behandelt.

Themen

- Überblick über die operativen Aufgaben von Amazon DocumentDB
- Überblick über die globalen Amazon DocumentDB-Cluster
- Verwaltung von Amazon DocumentDB-Clustern
- Verwaltung von Amazon DocumentDB DocumentDB-Instances
- Amazon DocumentDB-Subnetzgruppen verwalten
- Amazon DocumentDB Hochverfügbarkeit und Replikation
- Amazon DocumentDB DocumentDB-Indizes verwalten
- Verwaltung der Dokumentenkomprimierung auf Sammlungsebene
- Amazon DocumentDB DocumentDB-Ereignisse verwalten
- Auswahl von Regionen und Availability Zones
- Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen
- Grundlegendes zu Amazon DocumentDB DocumentDB-Endpunkten
- Grundlegendes zu Amazon DocumentDB Amazon Resource Names () ARNs
- Taggen von Amazon DocumentDB DocumentDB-Ressourcen
- Wartung von Amazon DocumentDB
- Grundlegendes zu Rollen im Zusammenhang mit Services
- Change-Streams mit Amazon DocumentDB verwenden
- Verwendung AWS Lambda mit Change-Streams

Überblick über die operativen Aufgaben von Amazon DocumentDB

Dieser Abschnitt behandelt betriebliche Aufgaben für Ihren Amazon DocumentDB-Cluster und wie Sie diese Aufgaben mit dem AWS CLI erledigen können.

Themen

- Hinzufügen eines Replikats zu einem Amazon DocumentDB-Cluster
- Beschreibung von Clustern und Instanzen
- Einen Cluster-Snapshot erstellen
- Wiederherstellung aus einem Snapshot
- Eine Instanz aus einem Cluster entfernen
- Löschen eines Clusters

Hinzufügen eines Replikats zu einem Amazon DocumentDB-Cluster

Nachdem Sie die primäre Instance für Ihren Amazon DocumentDB-Cluster erstellt haben, können Sie ein oder mehrere Replikate hinzufügen. Ein Replikat ist eine schreibgeschützte Instance, die zwei Zwecken dient:

- Skalierbarkeit — Wenn Sie über eine große Anzahl von Clients verfügen, die gleichzeitigen Zugriff benötigen, können Sie weitere Replikate für die Leseskalierung hinzufügen.
- Hochverfügbarkeit — Wenn die primäre Instance ausfällt, wechselt Amazon DocumentDB automatisch zu einer Replikat-Instance und bestimmt diese als neue primäre Instanz. Wenn ein Replikat ausfällt, können andere Instances im Cluster nach wie vor für die Bearbeitung von Anfragen verwendet werden, bis der ausgefallene Knoten wiederhergestellt werden kann.

Jeder Amazon DocumentDB-Cluster kann bis zu 15 Replikate unterstützen.

> **Note**
>
> Für maximale Fehlertoleranz sollten Sie Replikate in separaten Availability Zones bereitstellen. Dadurch wird sichergestellt, dass Ihr Amazon DocumentDB-Cluster auch dann weiter funktioniert, wenn eine gesamte Availability Zone nicht verfügbar ist.

Das folgende AWS CLI Beispiel zeigt, wie ein neues Replikat hinzugefügt wird. Der `--availability-zone`-Parameter platziert das Replikat in der angegebenen Availability Zone.

```
aws docdb create-db-instance \
    --db-instance-identifier sample-instance \
    --db-cluster-identifier sample-cluster \
```

```
    --engine docdb \
    --db-instance-class db.r5.large \
    --availability-zone us-east-1a
```

Beschreibung von Clustern und Instanzen

Das folgende AWS CLI Beispiel listet alle Amazon DocumentDB-Cluster in einer Region auf. Für bestimmte Verwaltungsfunktionen wie Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird. Der `filterName=engine,Values=docdb` Filterparameter gibt nur Amazon DocumentDB-Cluster zurück.

Weitere Informationen zum Beschreiben und Ändern von Clustern finden Sie im Amazon DocumentDB-Cluster-Lebenszyklus.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusters": [
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-1",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
```

```
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-2",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-3",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        }
    ]
}
```

Das folgende AWS CLI Beispiel listet die Instances in einem Amazon DocumentDB-Cluster auf. Weitere Informationen zum Beschreiben und Ändern von Clustern finden Sie im Lebenszyklus einer Amazon DocumentDB DocumentDB-Instanz.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Die Ausgabe sieht wie unten aus. Diese Ausgabe enthält zwei Instances. Die Primär-Instance ist `sample-instance-1` (`"IsClusterWriter": true`). Es gibt auch eine Replikat-Instance, nämlich `sample-instance2` (`"IsClusterWriter: false"`).

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
```

```
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-cluster-2",
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ]
    ]
]
```

Einen Cluster-Snapshot erstellen

Ein Cluster-Snapshot ist eine vollständige Sicherung der Daten in Ihrem Amazon DocumentDB-Cluster. Wenn der Snapshot erstellt wird, liest Amazon DocumentDB Ihre Daten direkt aus dem Cluster-Volume. Aus diesem Grund können Sie einen Snapshot erstellen, auch wenn zum jeweiligen Zeitpunkt keine Instances in Ihrem Cluster ausgeführt werden. Die zum Erstellen eines Snapshots erforderliche Zeit hängt von der Größe Ihres Cluster-Volumes ab.

Amazon DocumentDB unterstützt automatische Backups, die täglich während des bevorzugten Backup-Fensters — einem Zeitraum von 30 Minuten am Tag — durchgeführt werden. Das folgende AWS CLI Beispiel zeigt, wie Sie das Backup-Fenster für Ihren Cluster anzeigen können:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].PreferredBackupWindow'
```

Die Ausgabe zeigt das Sicherungszeitfenster (in UTC):

```
[
    "00:18-00:48"
]
```

Sie können das Backup-Fenster definieren, wenn Sie Ihren Amazon DocumentDB-Cluster erstellen. Sie können das Sicherungszeitfenster auch ändern, wie im folgenden Beispiel gezeigt wird. Wenn Sie kein Backup-Fenster definieren, weist Amazon DocumentDB Ihrem Cluster automatisch eines zu.

```
aws docdb modify-db-cluster \
```

```
    --db-cluster-identifier sample-cluster \
    --preferred-backup-window "02:00-02:30"
```

Zusätzlich zu automatischen Sicherungen können Sie jederzeit manuell einen Cluster-Snapshot erstellen. Wenn Sie dies tun, müssen Sie das zu sichernde Cluster festlegen und dem Snapshot einen eindeutigen Namen geben, damit er später für Wiederherstellungszwecke verwendet werden kann.

Das folgende AWS CLI Beispiel zeigt, wie Sie einen Snapshot Ihrer Daten erstellen.

```
aws docdb create-db-cluster-snapshot \
    --db-cluster-identifier sample-cluster \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

Wiederherstellung aus einem Snapshot

Sie können einen Cluster-Snapshot in einem neuen Amazon DocumentDB-Cluster wiederherstellen. Dazu geben Sie den Namen des Snapshots und den Namen eines neuen Clusters an. Sie können keine Wiederherstellung von einem Snapshot in einem vorhandenen Cluster durchführen. Stattdessen erstellt Amazon DocumentDB bei der Wiederherstellung einen neuen Cluster und füllt ihn dann mit Ihren Snapshot-Daten.

Das folgende Beispiel zeigt alle Snapshots für den Cluster `sample-cluster`.

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusterSnapshots[*].[DBClusterSnapshotIdentifier,SnapshotType,Status]'
```

Die Ausgabe sieht ungefähr wie folgt aus. Ein manueller Snapshot ist ein Snapshot, den Sie manuell erstellt haben, wohingegen ein automatisierter Snapshot von Amazon DocumentDB innerhalb des Cluster-Backup-Fensters erstellt wird.

```
[
        "sample-cluster-snapshot",
        "manual",
        "available"
    ],
    [
        "rds:sample-cluster",
        "automated",
```

```
        "available"
    ]
]
```

Das folgende Beispiel zeigt, wie ein Amazon DocumentDB-Cluster aus einem Snapshot wiederhergestellt wird.

```
aws docdb restore-db-cluster-from-snapshot \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster \
    --snapshot-identifier sample-cluster-snapshot
```

Dem neuen Cluster sind keine Instances zugeordnet. Wenn Sie mit dem Cluster interagieren möchten, müssen Sie ihm also eine Instance hinzufügen.

```
aws docdb create-db-instance \
    --db-instance-identifier new-sample-instance \
    --db-instance-class db.r5.large \
    --engine docdb \
    --db-cluster-identifier new-sample-cluster
```

Sie können die folgenden AWS CLI Operationen verwenden, um den Fortschritt der Cluster- und Instance-Erstellung zu überwachen. Wenn die Cluster- und Instance-Status verfügbar sind, können Sie eine Verbindung zum neuen Endpunkt des Clusters herstellen und auf Ihre Daten zugreifen.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier new-sample-cluster  \
    --query 'DBClusters[*].[Status,Endpoint]'
```

```
aws docdb describe-db-instances \
    --db-instance-identifier new-sample-instance \
    --query 'DBInstances[*].[DBInstanceStatus]'
```

Eine Instanz aus einem Cluster entfernen

Amazon DocumentDB speichert all Ihre Daten im Cluster-Volume. Die Daten bleiben in diesem Cluster-Volume erhalten. Dies gilt auch dann, wenn Sie alle Instances von Ihrem Cluster entfernen. Wenn Sie erneut auf die Daten zugreifen müssen, können Sie jederzeit dem Cluster eine Instance hinzufügen und da weitermachen, wo Sie aufgehört haben.

Das folgende Beispiel zeigt, wie Sie eine Instance aus Ihrem Amazon DocumentDB-Cluster entfernen.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-instance
```

Löschen eines Clusters

Bevor Sie einen Amazon DocumentDB-Cluster löschen können, müssen Sie zunächst alle zugehörigen Instances entfernen. Das folgende AWS CLI Beispiel gibt Informationen über die Instances in einem Cluster zurück. Wenn diese Operation Instance-Kennungen zurückgibt, müssen Sie die einzelnen Instances löschen. Weitere Informationen finden Sie unter Eine Instanz aus einem Cluster entfernen.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterMembers[*].DBInstanceIdentifier'
```

Wenn keine Instances mehr vorhanden sind, können Sie das Cluster löschen. Sie müssen dann eine der folgenden Optionen wählen:

- Einen endgültigen Snapshot erstellen — Erfassen Sie alle Clusterdaten in einem Snapshot, sodass Sie später eine neue Instanz mit diesen Daten neu erstellen können. Das Verfahren wird im folgenden Beispiel beschrieben:

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --final-db-snapshot-identifier sample-cluster-snapshot
  ```

- Den letzten Snapshot überspringen — Alle Clusterdaten werden dauerhaft gelöscht. Diese Aktion ist unwiderruflich. Das Verfahren wird im folgenden Beispiel beschrieben:

  ```
  aws docdb delete-db-cluster \
      --db-cluster-identifier sample-cluster \
      --skip-final-snapshot
  ```

Überblick über die globalen Amazon DocumentDB-Cluster

Was ist ein globaler Cluster?

Ein globaler Cluster besteht aus einer primären Region und bis zu fünf schreibgeschützten sekundären Regionen. Sie führen Schreibvorgänge direkt in den primären Cluster in der primären Region durch und Amazon DocumentDB repliziert die Daten mithilfe einer dedizierten Infrastruktur automatisch in die sekundären Regionen. Die Latenz liegt in der Regel unter einer Sekunde.

Wie sind globale Cluster nützlich?

- Wiederherstellung nach regionalen Ausfällen — Im Falle eines regionsweiten Ausfalls können Sie einen der sekundären Cluster innerhalb von Minuten zu einem primären Cluster heraufstufen, wobei ein typisches Recovery Time Objective (RTO) unter einer Minute liegt. Das Recovery Point Objective (RPO) wird in der Regel in Sekunden gemessen. Dies hängt jedoch von der Verzögerung im Netzwerk zum Zeitpunkt des Ausfalls ab.
- Globale Lesevorgänge mit lokaler Latenz — Wenn Sie Niederlassungen auf der ganzen Welt haben, können Sie einen globalen Cluster verwenden, um Ihre wichtigsten Informationsquellen in der Primärregion auf dem neuesten Stand zu halten. Niederlassungen in Ihren anderen Regionen können mit lokaler Latenz auf die Informationen in ihrer eigenen Region zugreifen.
- Skalierbare sekundäre Cluster — Sie können Ihre sekundären Cluster skalieren, indem Sie einer sekundären Region weitere Nur-Lese-Instances hinzufügen. Der sekundäre Cluster ist schreibgeschützt, sodass er bis zu 16 schreibgeschützte Replikatinstanzen anstelle der üblichen Grenze von 15 für einen einzelnen Cluster unterstützen kann.
- Schnelle Replikation vom primären zum sekundären Cluster — Die von einem globalen Cluster durchgeführte Replikation hat nur geringe Auswirkungen auf die Leistung des primären Datenbank-Clusters. Die Ressourcen der DB-Instances werden ausschließlich für Lese- und Schreib-Workloads von Anwendungen genutzt.

Was sind die aktuellen Einschränkungen globaler Cluster?

- Globale Cluster werden in Amazon DocumentDB v3.6 nicht unterstützt.
- Globale Cluster werden auf den Instance-Typen t3, t4g und r4 nicht unterstützt.
- Globale Cluster sind in den folgenden Regionen nicht verfügbar: Südamerika (São Paulo), Europa (Mailand), China (Peking) und China (Ningxia).

- Switchover und globales Failover werden nicht unterstützt, wenn Regionen unterschiedliche Engine-Versionen verwenden. Manuelles Failover wird unterstützt, wenn die Engine-Versionen nicht übereinstimmen.
- Nur der primäre Cluster führt Schreibvorgänge aus. Clients, die Schreibvorgänge ausführen, stellen eine Verbindung zum Clusterendpunkt des primären Clusters her.
- Sie können maximal fünf sekundäre Regionen und eine primäre Region für Ihren Cluster einrichten.
- Ein sekundärer Cluster kann nicht gestoppt werden. Ein primärer Cluster kann nicht gestoppt werden, wenn ihm sekundäre Cluster zugeordnet sind. Nur ein regionaler Cluster ohne sekundäre Cluster kann gestoppt werden.
- An den sekundären Cluster angehängte Replikate können unter bestimmten Umständen neu gestartet werden. Wenn die Instanz der primären Region neu gestartet oder ein Failover durchgeführt wird, werden auch die Replikate in der sekundären Region neu gestartet. Der Cluster ist dann nicht verfügbar, bis alle Replikate wieder mit der Writer-Instance des primären Datenbankclusters synchronisiert sind. Dieses Verhalten wird erwartet. Stellen Sie sicher, dass Sie die Auswirkungen auf Ihren globalen Cluster verstehen, bevor Sie Änderungen an Ihrem primären Cluster vornehmen.
- Sie können Change-Streams nicht auf sekundären Clustern verwenden.

Themen

- Schnellstartanleitung: globale Cluster
- Verwaltung eines globalen Amazon DocumentDB-Clusters
- Stellen Sie eine Connect zu einem globalen Amazon DocumentDB-Cluster her
- Überwachung globaler Amazon DocumentDB-Cluster
- Disaster Recovery und globale Amazon DocumentDB-Cluster

Schnellstartanleitung: globale Cluster

Themen

- Konfiguration
- Einen globalen Amazon DocumentDB-Cluster erstellen
- Hinzufügen eines AWS-Region zu einem globalen Amazon DocumentDB-Cluster
- Verwenden eines Snapshots für Ihren globalen Amazon DocumentDB-Cluster

Konfiguration

Der globale Amazon DocumentDB-Cluster umfasst mindestens zwei. AWS-Regionen Die primäre Region unterstützt einen Cluster mit einer primären (Writer-) Instance und bis zu 15 Replikat-Instances, während in einer sekundären Region ein schreibgeschütztes Cluster läuft, das ausschließlich aus bis zu 16 Replikat-Instances besteht. Ein globaler Cluster kann bis zu fünf sekundäre Regionen haben. In der Tabelle sind die maximal zulässigen Cluster, Instances und Replikate in einem globalen Cluster aufgeführt.

Beschreibung	Primär AWS-Region	Sekundär AWS-Region
Cluster	1	5 (maximal)
Writer-Inst	1	0
Nur-Lese-Instances (Amazon DocumentDB DocumentDB-Repliken) pro Cluster	15 (Max.)	16 (Total)
Schreibgeschützte Instances (maximal zulässig, bei tatsächlicher Anzahl von sekundären Regionen)	15 - s	s = Gesamtzahl der sekundäre n AWS-Regionen

Für die Cluster gelten die folgenden spezifischen Anforderungen:

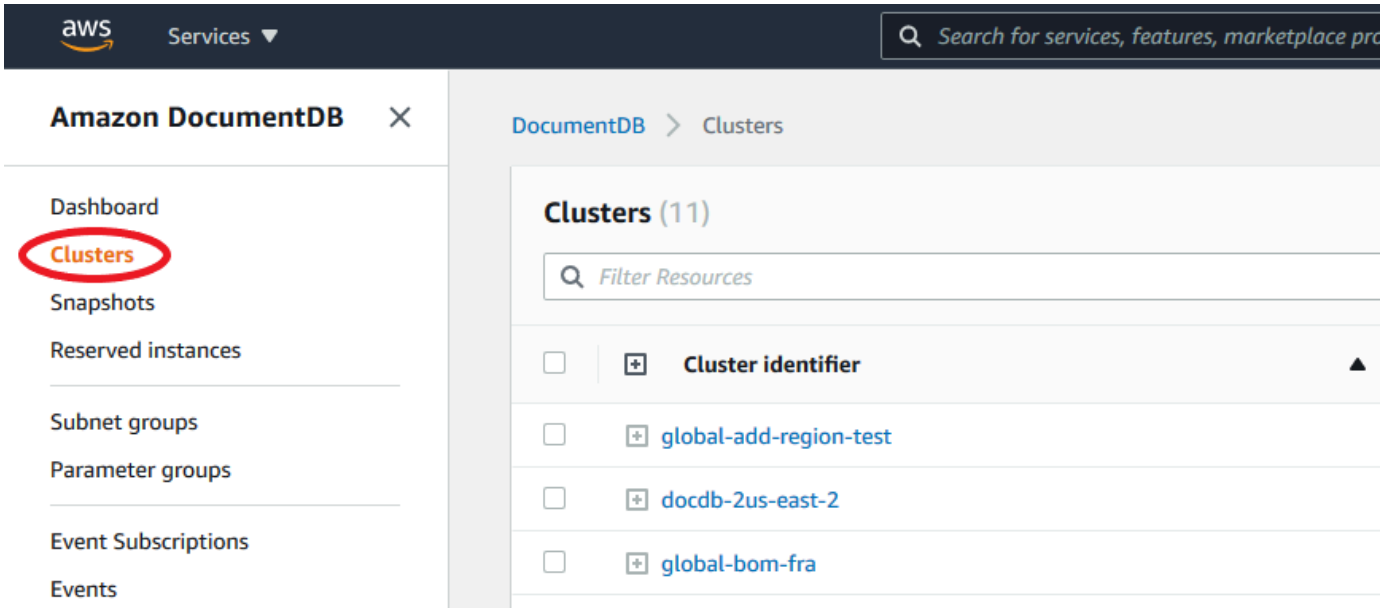
- Klassenanforderungen für Datenbank-Instances — Sie können nur die Klassen `db.r5` und `db.r6g` Instance verwenden.
- AWS-Region Anforderungen — Der primäre Cluster muss sich in einer Region befinden, und mindestens ein sekundärer Cluster muss sich in einer anderen Region desselben Kontos befinden. Sie können bis zu fünf sekundäre (schreibgeschützte) Cluster erstellen, und jeder muss sich in einer anderen Region befinden. Mit anderen Worten, es können sich keine zwei Cluster in derselben Region befinden.
- Anforderungen an die Benennung — Die Namen, die Sie für jeden Ihrer Cluster wählen, müssen in allen Regionen eindeutig sein. Sie können nicht denselben Namen für verschiedene Cluster verwenden, auch wenn sie sich in verschiedenen Regionen befinden.

Einen globalen Amazon DocumentDB-Cluster erstellen

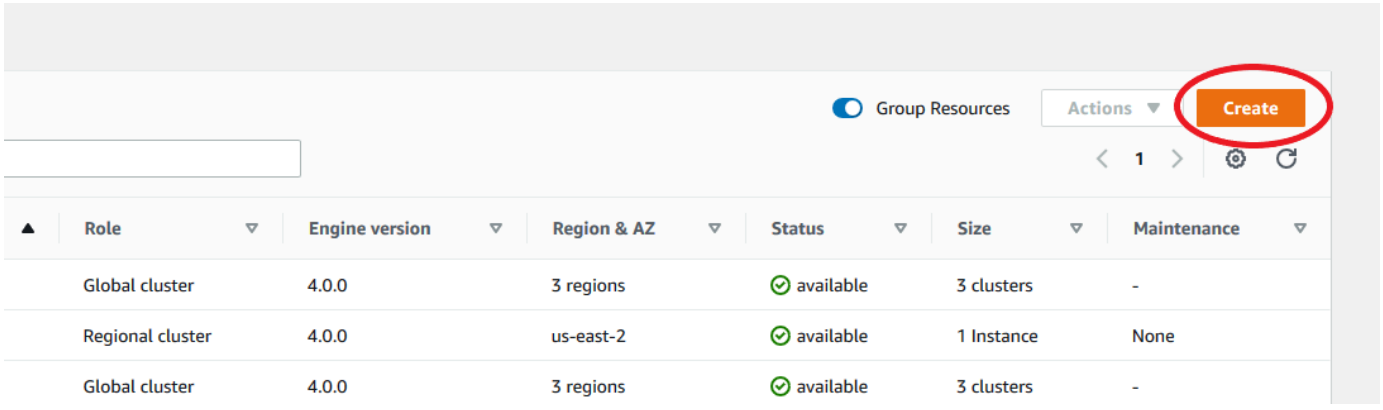
Sind Sie bereit, Ihren ersten globalen Cluster aufzubauen? In diesem Abschnitt erklären wir, wie Sie einen brandneuen globalen Cluster mit neuen Datenbankclustern und Instances erstellen, indem Sie entweder die AWS Management Console oder AWS CLI die folgenden Anweisungen verwenden.

Mit dem AWS Management Console

1. Navigieren Sie in der AWS Management Console zu Amazon DocumentDB.
2. Wenn Sie zur Amazon DocumentDB DocumentDB-Konsole gelangen, wählen Sie Clusters.

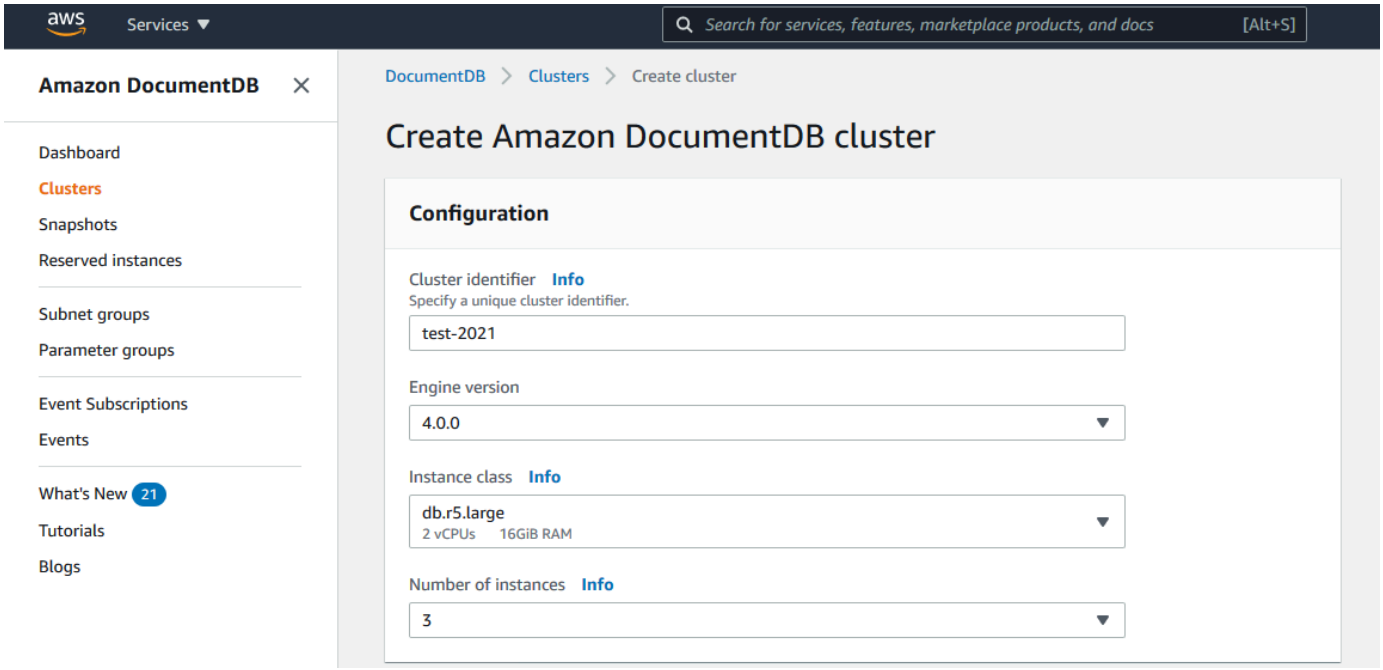


3. Wählen Sie Create (Erstellen) aus.

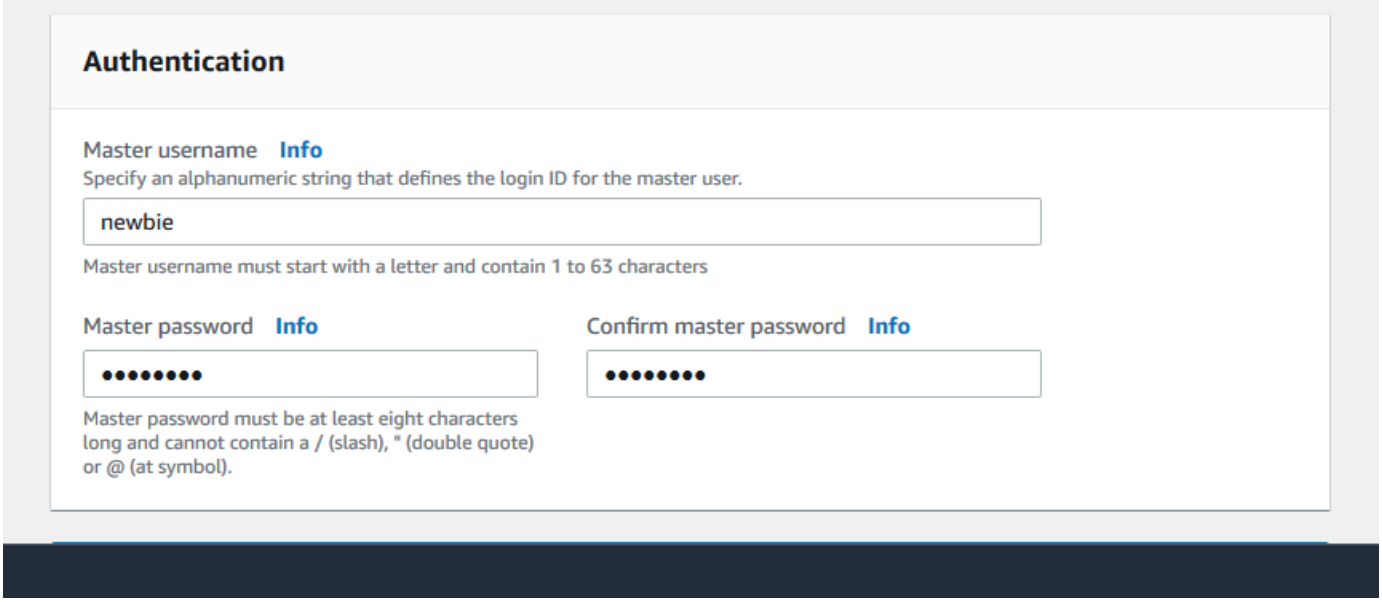


4. Füllen Sie den Abschnitt Konfiguration des Formulars Amazon DocumentDB-Cluster erstellen entsprechend aus:

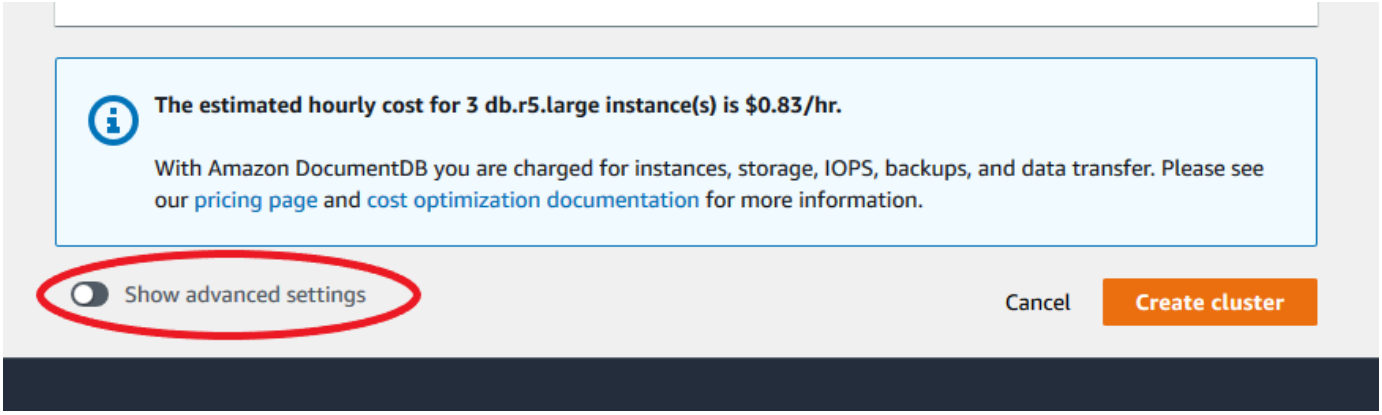
- Cluster-ID: Sie können entweder eine eindeutige Kennung für diese Instance eingeben oder Amazon DocumentDB erlauben, die Instance-ID auf der Grundlage der Cluster-ID bereitzustellen.
- Engine-Version: Wählen Sie 4.0.0
- Instanzklasse: Wählen Sie db.r5.large
- Anzahl der Instanzen: Wählen Sie 3.



5. Geben Sie im Abschnitt Authentifizierung einen Master-Benutzernamen und ein Master-Passwort ein.

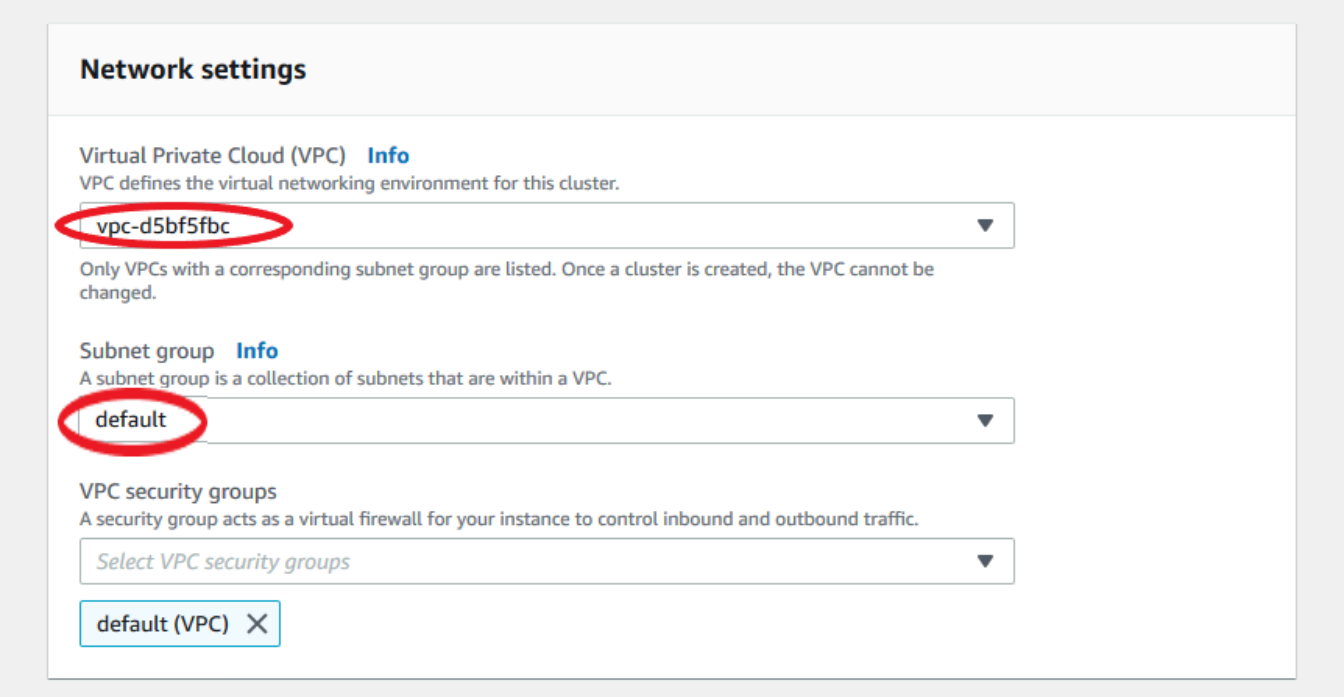


6. Wählen Sie Erweiterte Einstellungen anzeigen.

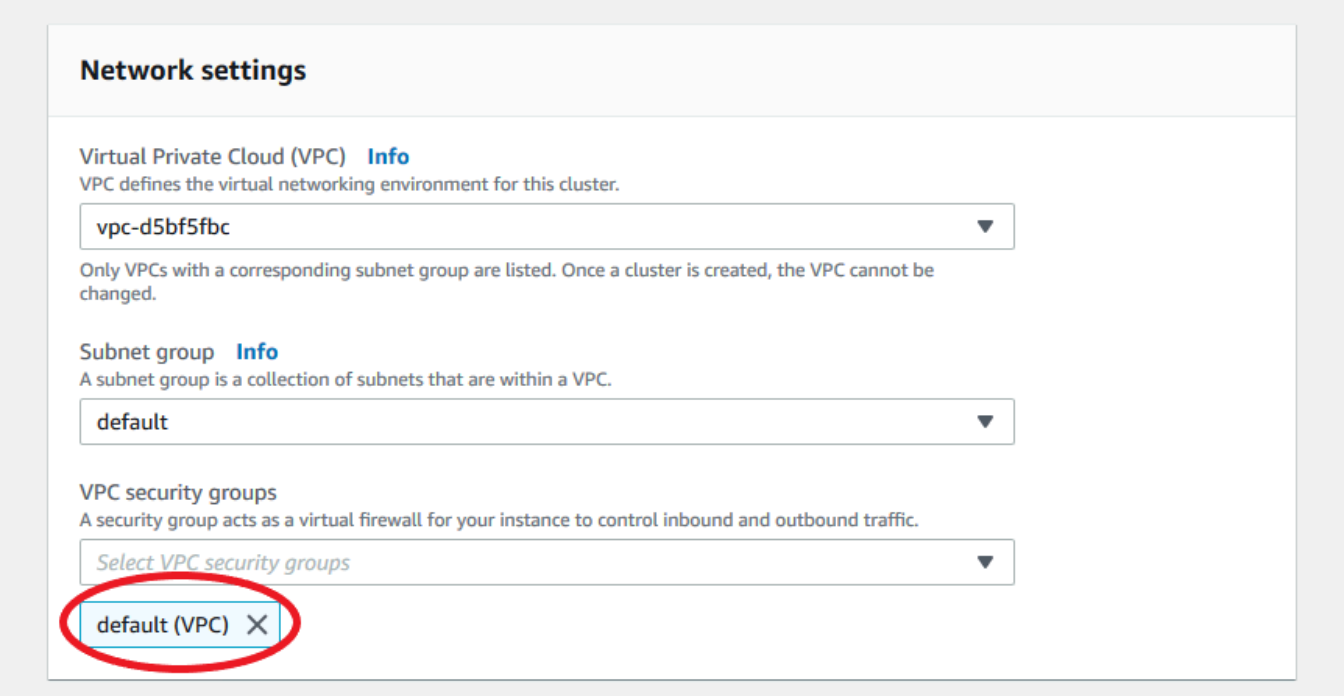


7. Gehen Sie im Bereich Netzwerkeinstellungen wie folgt vor:

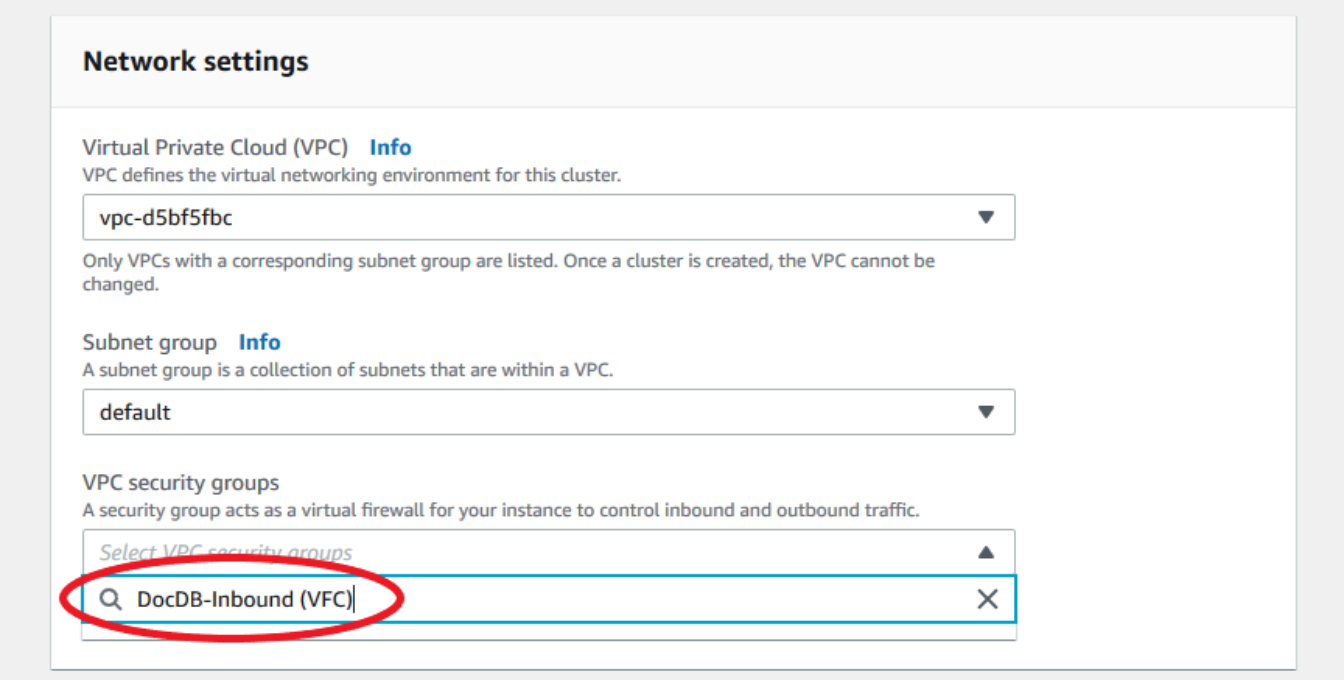
   - Behalten Sie die Standardoptionen für Virtual Private Cloud (VPC) und Subnetzgruppe bei.



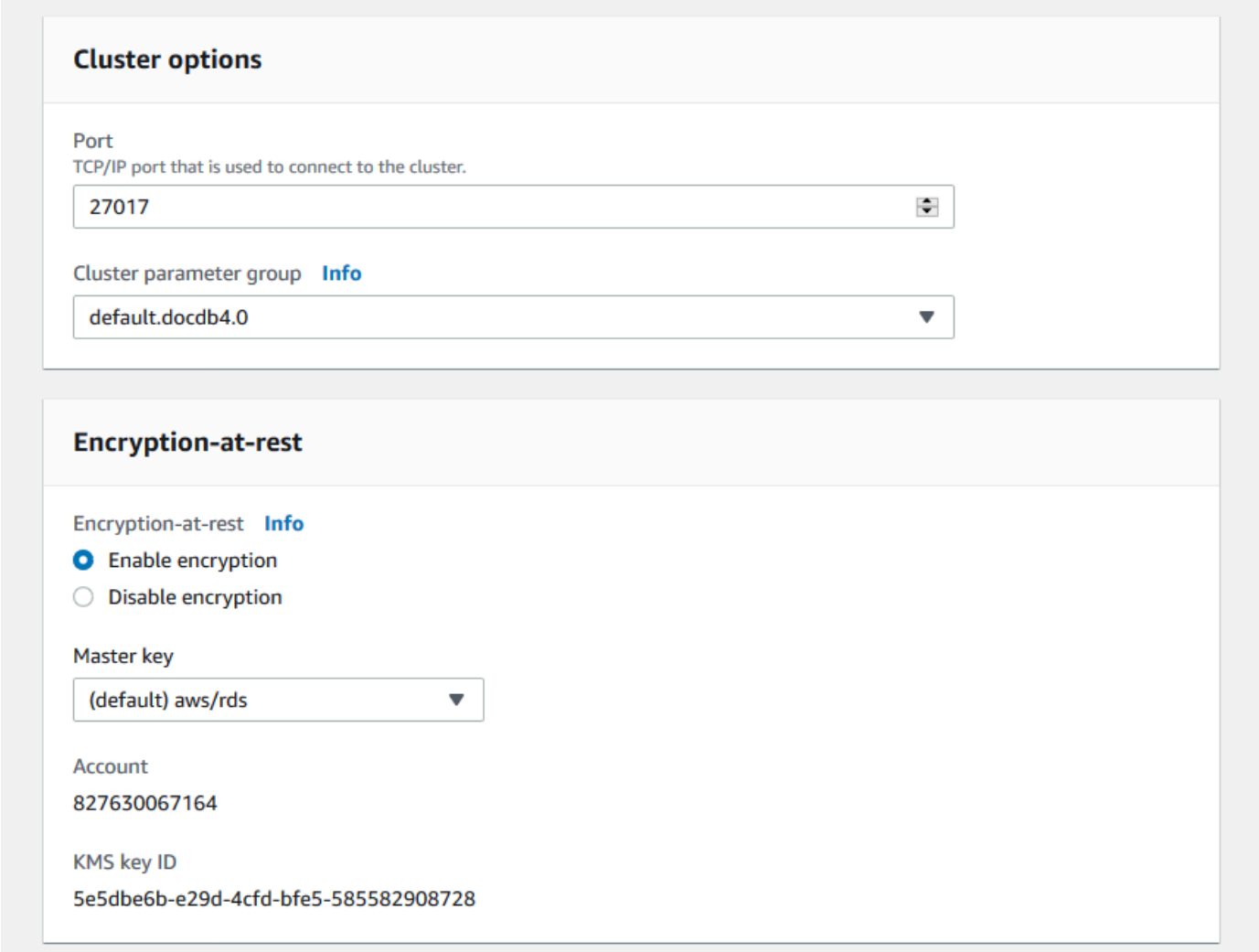
- Für VPC-Sicherheitsgruppen sollte Standard (VPC) bereits hinzugefügt sein.



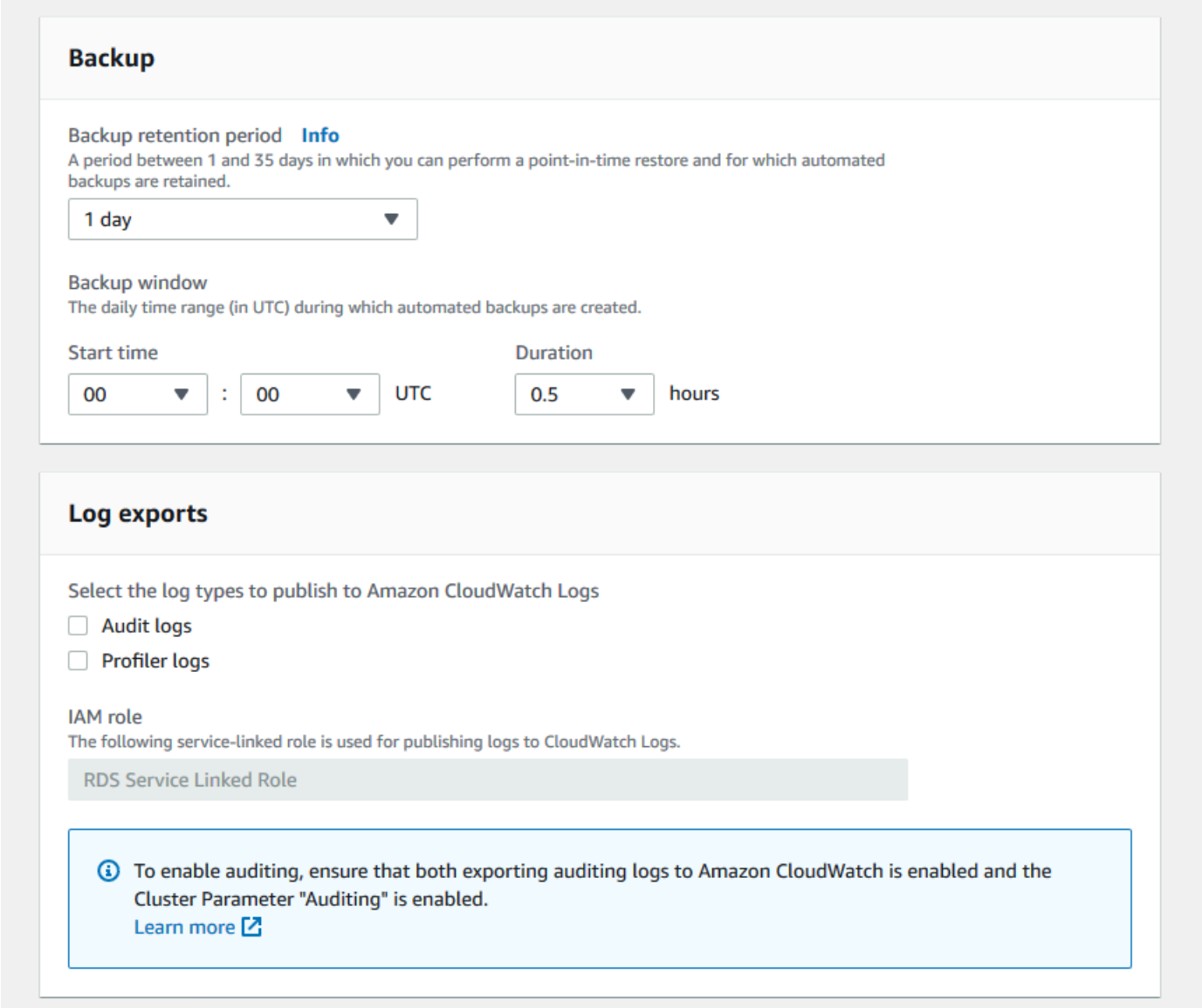
- Geben Sie `DocDB` in das Feld VPC-Sicherheitsgruppen ein und wählen Sie DoCDB-Inbound (VFC) aus.



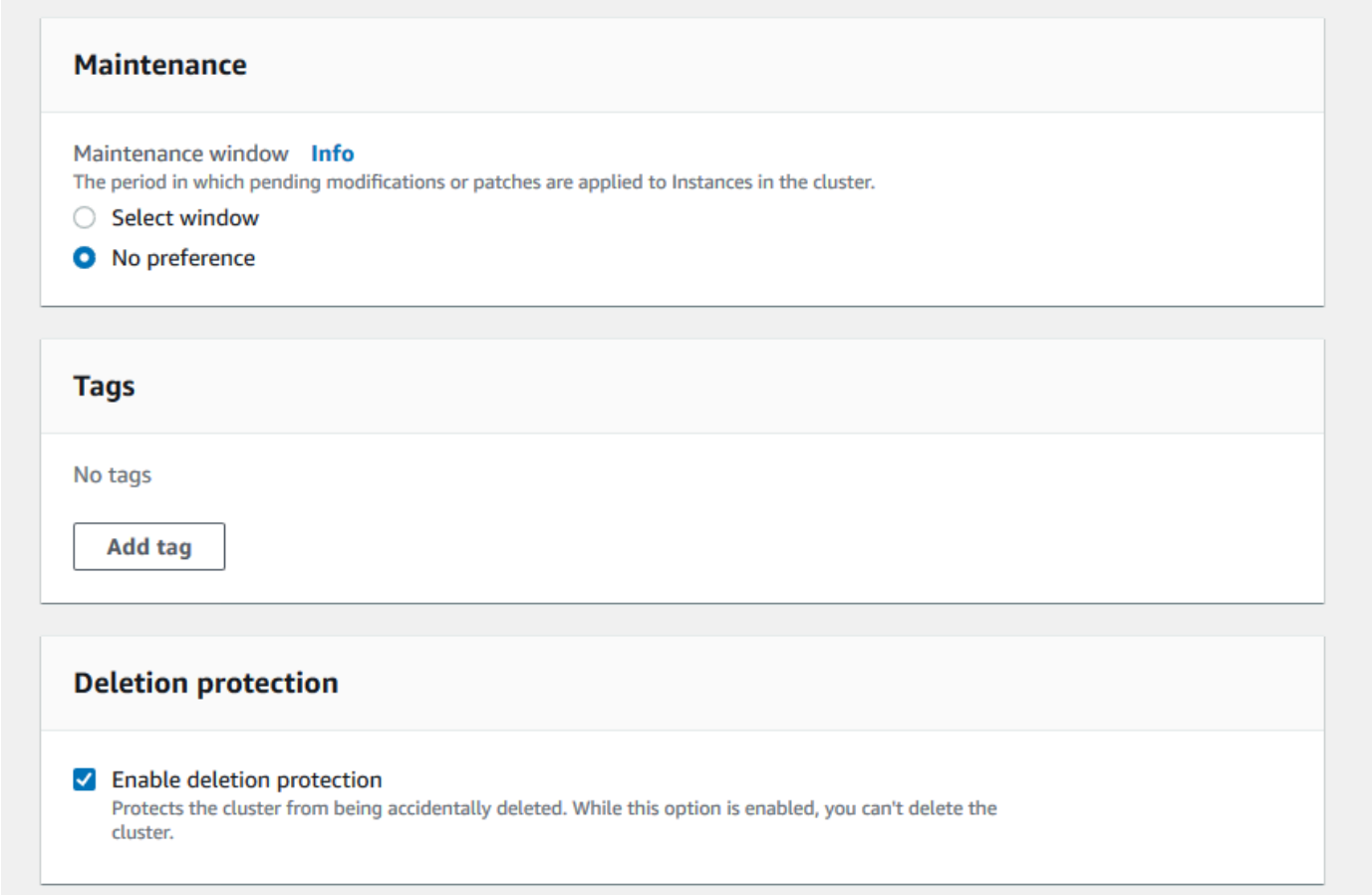
8. Behalten Sie für Cluster-Optionen und E die Standardauswahl bei. ncryption-at-rest



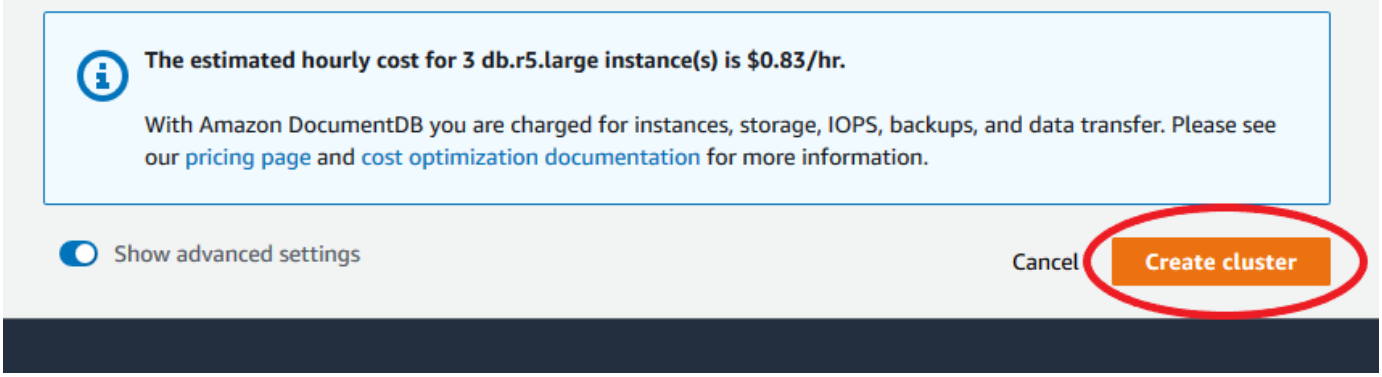
9. Behalten Sie für Backup - und Protokollexporte die Standardauswahl bei.



10. Behalten Sie für Wartung, Tags und Löschschutz die Standardauswahl bei.



11. Klicken Sie nun auf die Schaltfläche mit der Aufschrift Cluster erstellen.



Mit dem AWS CLI

Um einen regionalen Amazon DocumentDB DocumentDB-Cluster zu erstellen, rufen Sie den auf create-global-cluster AWS CLI. Der folgende AWS CLI Befehl erstellt einen Amazon DocumentDB-Cluster mit dem Namen`global-cluster-id`. Weitere Informationen zum Löschschutz finden Sie unterLöschen eines Amazon DocumentDB-Clusters.

Außerdem `--engine-version` ist dies ein optionaler Parameter, der standardmäßig die neueste Hauptversion der Engine verwendet. Die aktuelle Version der Hauptengine ist`5.0.0`. Wenn neue Hauptversionen der Engine veröffentlicht werden, `--engine-version` wird die Standard-Engine-Version für aktualisiert, sodass sie der letzten Hauptversion der Engine entspricht. Daher empfehlen wir für Produktionsworkloads, insbesondere solche, die von Skripten, Automatisierung oder AWS CloudFormation Vorlagen abhängig sind, die Angabe der `--engine-version` beabsichtigten Hauptversion ausdrücklich.

Wenn ein `db-subnet-group-name` oder nicht angegeben `vpc-security-group-id` ist, verwendet Amazon DocumentDB die Standard-Subnetzgruppe und die Amazon VPC-Sicherheitsgruppe für die angegebene Region.

Ersetzen Sie im folgenden Beispiel jede Information durch Ihre *`user input placeholder`* eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --global-cluster-identifier global-cluster-id \
      --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

Für Windows:

```
aws docdb create-db-cluster ^
      --global-cluster-identifier global-cluster-id ^
      --source-db-cluster-identifier arn:aws:rds:us-east-1:111122223333:cluster-id
```

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

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "4.0.0",
```

```
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:cluster-id",
        "DBClusterIdentifier": "cluster-id",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb4.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "cluster-id.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "cluster-id.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

Die Erstellung des Clusters dauert mehrere Minuten. Sie können das AWS Management Console oder verwenden AWS CLI , um den Status Ihres Clusters zu überwachen. Weitere Informationen finden Sie unter Den Status eines Amazon DocumentDB-Clusters überwachen.

> **Important**
>
> Wenn Sie den verwenden AWS CLI , um einen regionalen Amazon DocumentDB-Cluster zu erstellen, werden keine Instances erstellt. Daher müssen Sie explizit eine primäre Instance und alle benötigten Replikat-Instances anlegen. Sie können entweder die Konsole oder AWS CLI zum Erstellen der Instances verwenden. Weitere Informationen finden Sie unter

Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster und CreateDBCluster in der Amazon DocumentDB DocumentDB-API-Referenz.

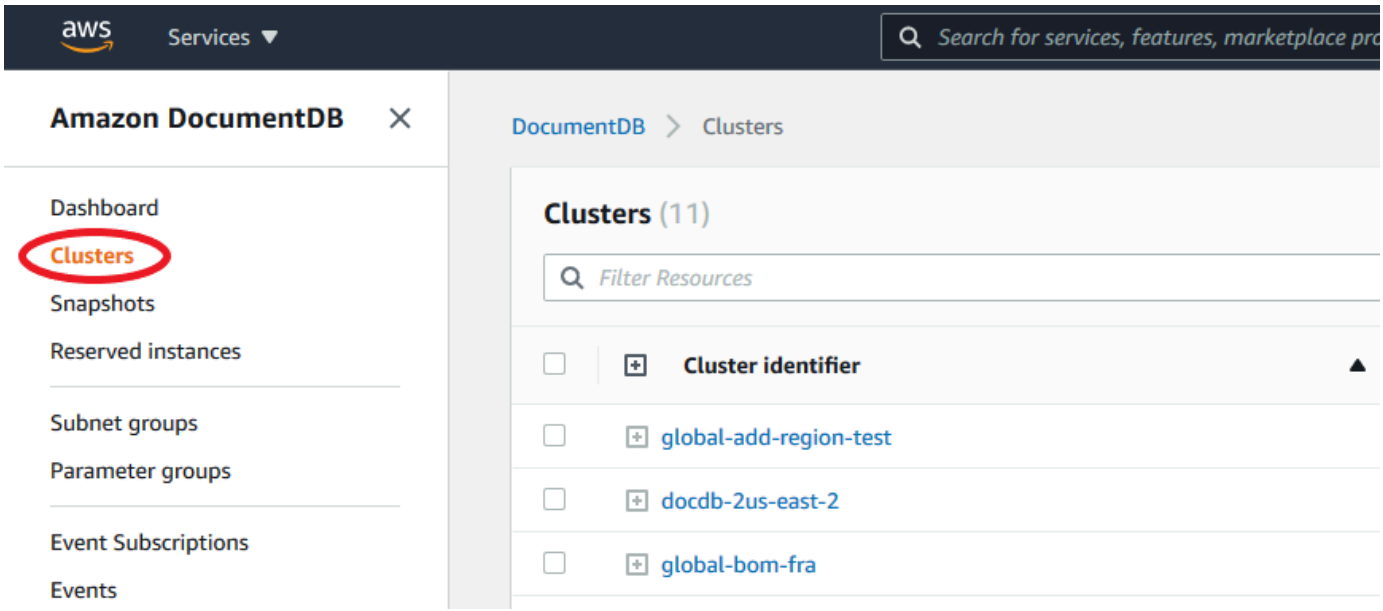
Sobald Ihr regionaler Cluster verfügbar ist, können Sie mit den folgenden Anweisungen einen sekundären Cluster in einer anderen Region hinzufügen:Hinzufügen eines AWS-Region zu einem globalen Amazon DocumentDB-Cluster. Wenn Sie eine Region hinzufügen, wird Ihr regionaler Cluster zu Ihrem primären Cluster, und Sie haben einen neuen sekundären Cluster in der von Ihnen ausgewählten Region.

Hinzufügen eines AWS-Region zu einem globalen Amazon DocumentDB-Cluster

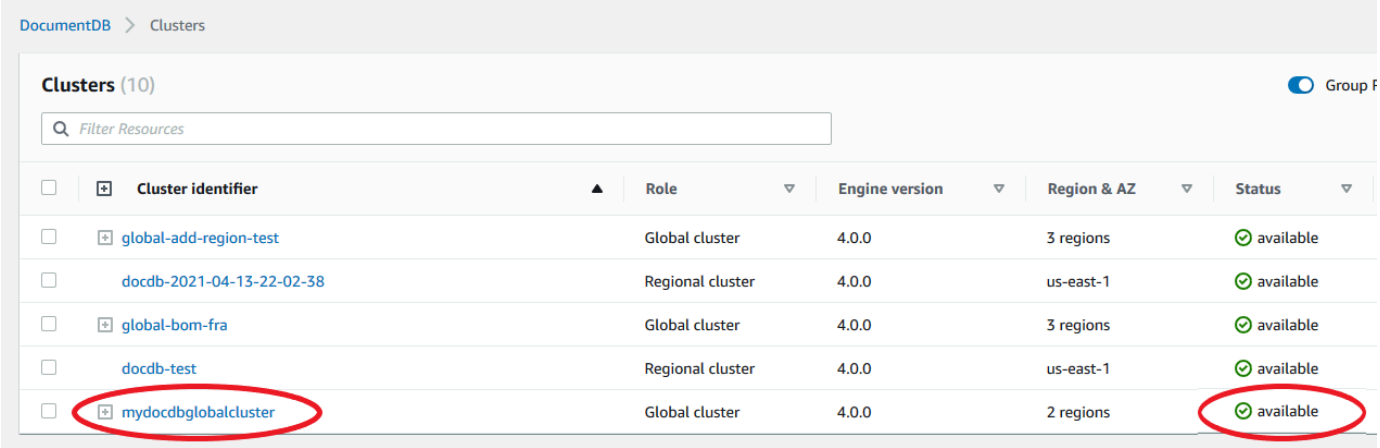
Ein globaler Cluster benötigt mindestens einen sekundären Cluster in einer anderen Region als der primäre Cluster, und Sie können bis zu fünf sekundäre Cluster hinzufügen. Beachten Sie, dass Sie für jeden sekundären Cluster, den Sie hinzufügen, die Anzahl der im primären Cluster zulässigen Replikate um eins reduzieren müssen. Wenn Ihr globaler Cluster beispielsweise fünf sekundäre Regionen hat, kann Ihr primärer Cluster nur 10 (statt 15) Replikate haben. Weitere Informationen finden Sie unter Konfigurationsanforderungen eines globalen Amazon DocumentDB-Clusters.

Mit dem AWS Management Console

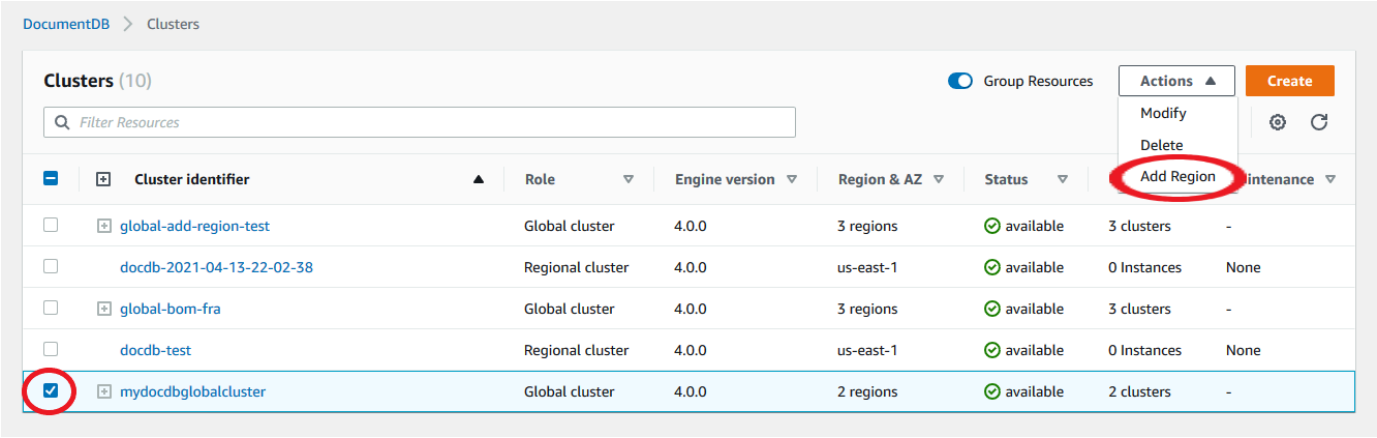
1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.



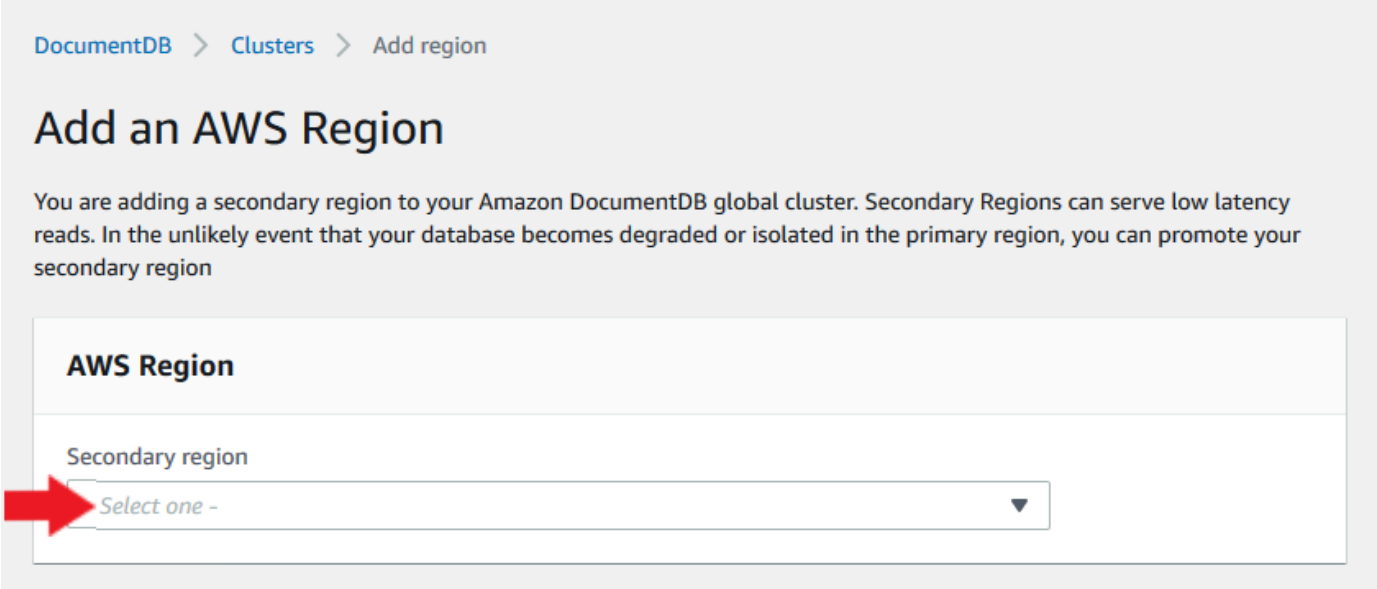
3. Wählen Sie den Cluster aus, dem Sie einen sekundären Cluster hinzufügen möchten. Stellen Sie sicher, dass der `Available` Cluster



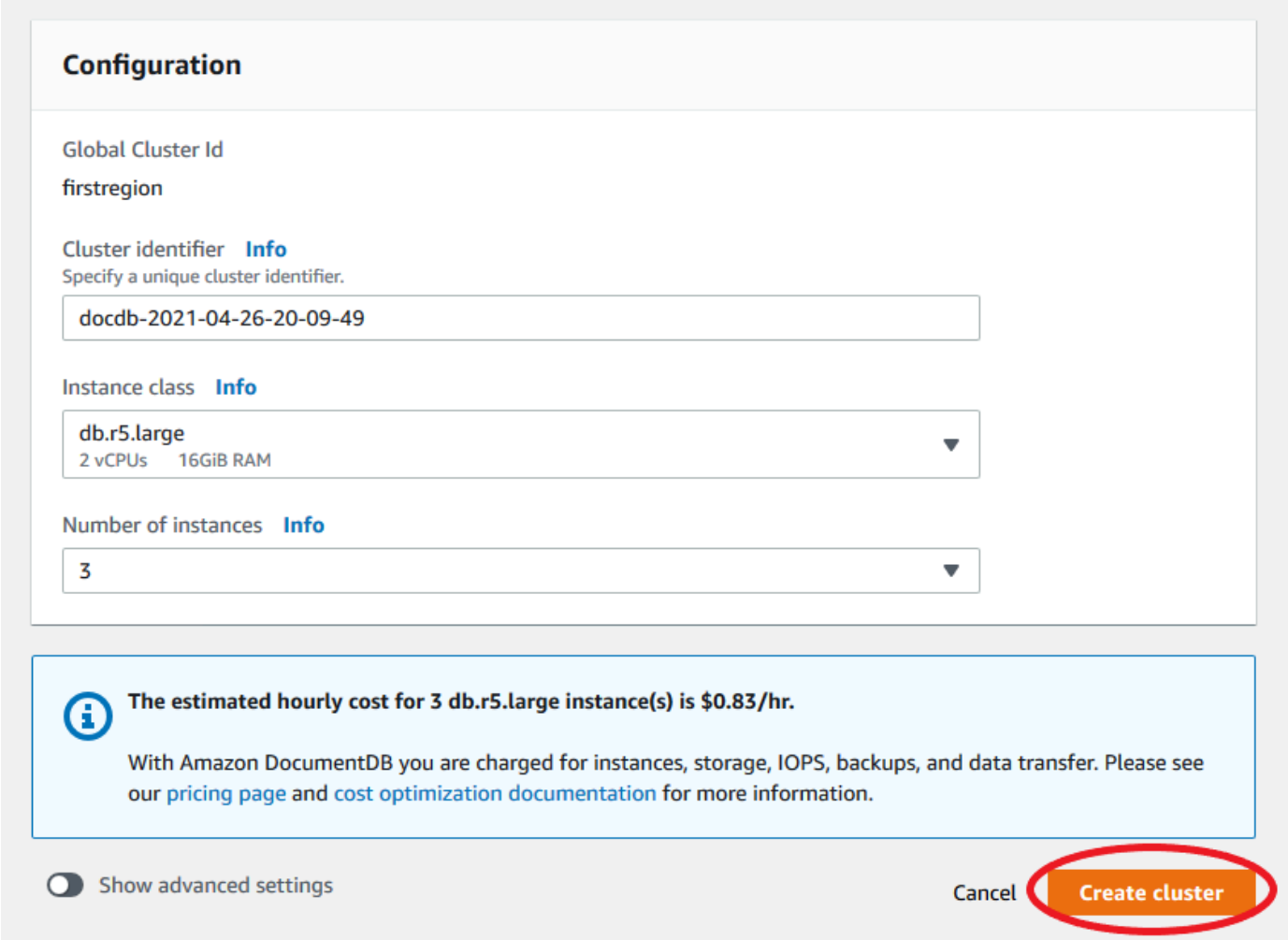
4. Wählen Sie die Dropdownliste für Aktionen aus und wählen Sie dann Region hinzufügen aus.



5. Wählen Sie auf der AWS-Region Seite Hinzufügen die sekundäre Region aus. Beachten Sie, dass Sie keine Region auswählen können, die bereits über einen sekundären Cluster für denselben globalen Cluster verfügt. Außerdem kann es sich nicht um dieselbe Region wie der primäre Cluster handeln. Wenn dies die erste Region ist, die Sie hinzufügen, müssen Sie auch eine globale Cluster-ID Ihrer Wahl angeben.



6. Füllen Sie die verbleibenden Felder für den sekundären Cluster in der neuen Region aus und wählen Sie dann Cluster erstellen aus. Nachdem Sie die Region hinzugefügt haben, können Sie sie in der Liste der Cluster in der sehen AWS Management Console.



Mit dem AWS CLI

- Verwenden Sie den `create-db-cluster` CLI-Befehl mit dem Namen (`--global-cluster-identifier`) Ihres globalen Clusters. Für andere Parameter, führen Sie die folgenden Schritte aus:
  - Wählen Sie für `--region` eine andere Region AWS-Region als die Ihrer Hauptregion.
  - Wählen Sie bestimmte Werte für die Parameter `--engine-version` und `--engine` aus.
  - Geben Sie für einen verschlüsselten Cluster Ihren primären Cluster AWS-Region als `--source-region` für die Verschlüsselung an.

Im folgenden Beispiel wird ein neuer Amazon DocumentDB-Cluster erstellt und als schreibgeschützter sekundärer Cluster an den globalen Cluster angehängt. Im letzten Schritt wird die Instance dem neuen Cluster hinzugefügt.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb --region secondary-region-id \
  create-db-cluster \
    --db-cluster-identifier cluster-id \
    --global-cluster-identifier global-cluster-id \
    --engine-version version \
    --engine docdb

aws docdb --region secondary-region-id \
  create-db-instance \
    --db-cluster-identifier cluster-id \
    --global-cluster-identifier global-cluster-id \
    --engine-version version \
    --engine docdb
```

Für Windows:

```
aws docdb --region secondary-region-id ^
  create-db-cluster ^
    --db-cluster-identifier cluster-id ^
    --global-cluster-identifier global-cluster-id ^
    --engine-version version ^
    --engine docdb

aws docdb --region secondary-region-id ^
  create-db-instance ^
    --db-cluster-identifier cluster-id ^
    --global-cluster-identifier global-cluster-id ^
    --engine-version version ^
    --engine docdb
```

Verwenden eines Snapshots für Ihren globalen Amazon DocumentDB-Cluster

Sie können einen Snapshot eines Amazon DocumentDB-Clusters wiederherstellen, um ihn als Ausgangspunkt für Ihren globalen Cluster zu verwenden. Dazu müssen Sie den Snapshot wiederherstellen und einen neuen Cluster erstellen. Dieser wird als primärer Cluster Ihres globalen

Clusters dienen. Anschließend können Sie dem wiederhergestellten Cluster eine weitere Region hinzufügen und ihn so in einen globalen Cluster konvertieren.

Verwaltung eines globalen Amazon DocumentDB-Clusters

Sie führen die meisten Verwaltungsvorgänge auf den einzelnen Clustern aus, aus denen ein globaler Cluster besteht. Wenn Sie auf der Seite Cluster in der Konsole die Option Gruppenverwandte Ressourcen auswählen, werden der primäre Cluster und die sekundären Cluster unter dem zugehörigen globalen Cluster gruppiert angezeigt.

Auf der Registerkarte Konfiguration für einen globalen Cluster werden AWS-Regionen der Standort der Cluster, die Version und die globale Cluster-ID angezeigt.

Themen

- Ändern eines globalen Amazon DocumentDB-Clusters
- Parameter eines globalen Amazon DocumentDB-Clusters ändern
- Einen Cluster aus einem globalen Amazon DocumentDB-Cluster entfernen
- Löschen eines Clusters aus einem globalen Amazon DocumentDB-Cluster
- Erstellen eines Headless-Amazon-DocumentDB-Clusters in einer sekundären Region

Ändern eines globalen Amazon DocumentDB-Clusters

Auf der Cluster-Seite AWS Management Console werden alle Ihre globalen Cluster aufgeführt, wobei der primäre Cluster und die sekundären Cluster für jeden Cluster angezeigt werden. Der globale Cluster hat seine eigenen Konfigurationseinstellungen. Insbesondere gibt es Regionen, die seinen primären und sekundären Clustern zugeordnet sind.

Wenn Sie Änderungen am globalen Cluster vornehmen, haben Sie die Möglichkeit, Änderungen rückgängig zu machen.

Wenn Sie Weiter wählen, bestätigen Sie die Änderungen.

Parameter eines globalen Amazon DocumentDB-Clusters ändern

Sie können die Cluster-Parametergruppen unabhängig für jeden Cluster innerhalb des globalen Clusters konfigurieren. Die meisten Parameter funktionieren genauso wie für andere Arten von Amazon DocumentDB-Clustern. Wir empfehlen, dass Sie die Einstellungen zwischen allen Clustern in einer globalen Datenbank konsistent halten. Dies hilft, unerwartete Verhaltensänderungen zu vermeiden, wenn Sie einen sekundären Cluster zum primären Cluster hochstufen.

Sie sollten z. B. die gleichen Einstellungen für Zeitzonen und Zeichensätze verwenden, um inkonsistentes Verhalten zu vermeiden, wenn ein anderer Cluster die Rolle des primären Clusters übernimmt.

Einen Cluster aus einem globalen Amazon DocumentDB-Cluster entfernen

Es gibt mehrere Situationen, in denen Sie möglicherweise Cluster aus Ihrem globalen Cluster entfernen möchten. Beispielsweise möchten Sie möglicherweise einen Cluster aus einem globalen Cluster entfernen, wenn der primäre Cluster heruntergestuft oder isoliert wird. Er wird dann zu einem eigenständigen bereitgestellten Cluster, der zur Erstellung eines neuen globalen Clusters verwendet werden kann. Weitere Informationen hierzu finden Sie unter Durchführen eines manuellen Failovers für einen globalen Amazon DocumentDB-Cluster.

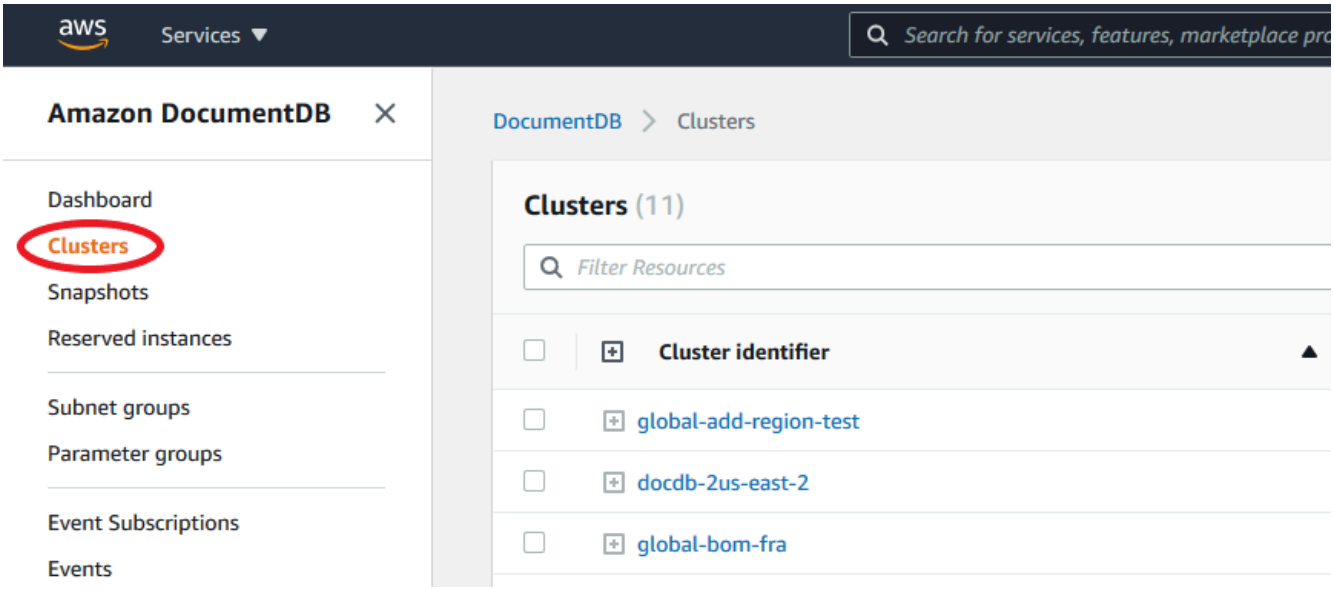
Möglicherweise möchten Sie auch Cluster entfernen, weil Sie einen globalen Cluster löschen möchten, den Sie nicht mehr benötigen. Sie können den globalen Cluster erst löschen, nachdem Sie alle zugehörigen Cluster getrennt haben und den primären Cluster als letzten übrig lassen. Weitere Informationen finden Sie unter Löschen eines Clusters aus einem globalen Amazon DocumentDB-Cluster.

> **Note**
>
> Wenn ein Cluster vom globalen Cluster getrennt wird, wird er nicht mehr mit dem primären Cluster synchronisiert. Er wird zu einem eigenständigen bereitgestellten Cluster mit vollen Lese-/Schreibfunktionen. Außerdem ist es in der Amazon DocumentDB DocumentDB-Konsole nicht mehr sichtbar. Es ist nur sichtbar, wenn Sie die Region in der Konsole auswählen, in der sich der Cluster befand.

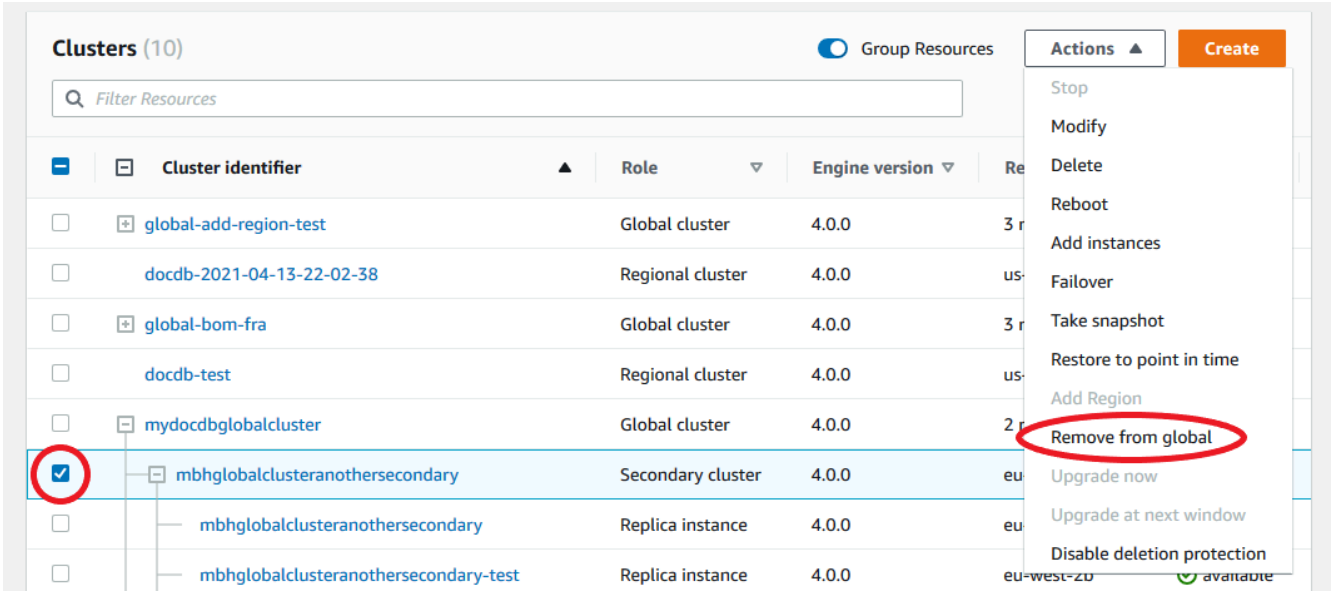
Sie können Cluster mithilfe der AWS Management Console, der oder der RDS-API aus Ihrem globalen Cluster entfernen. AWS CLI

Using the AWS Management Console

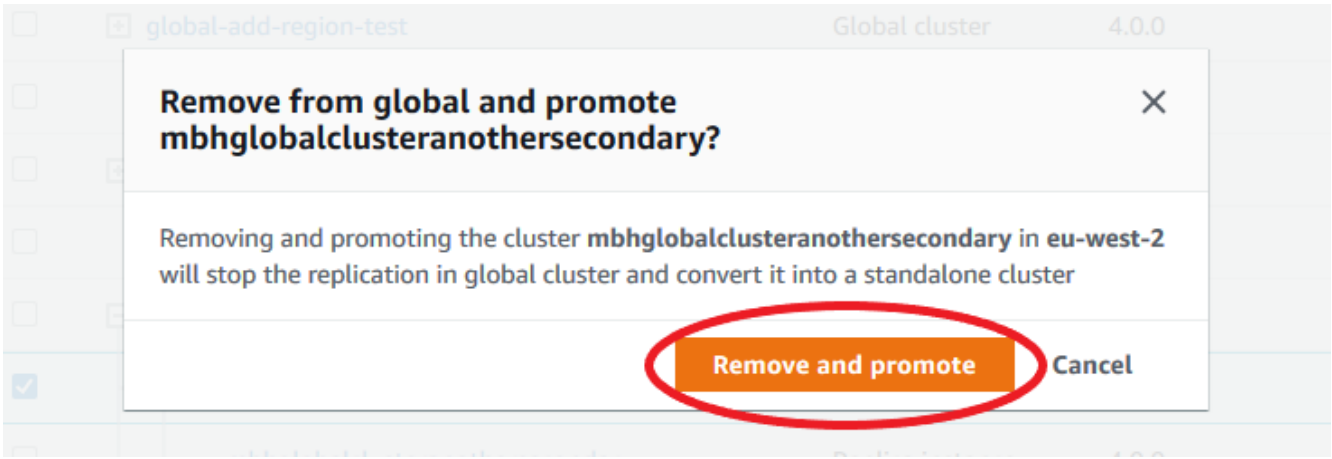
1. Melden Sie sich bei der an AWS Management Console und navigieren Sie zur Amazon DocumentDB DocumentDB-Konsole.
2. Wählen Sie in der linken Navigationsleiste Clusters aus.



3. Erweitern Sie den globalen Cluster, sodass Sie alle sekundären Cluster sehen können. Wählen Sie die sekundären Cluster aus, die Sie entfernen möchten. Wählen Sie Aktionen und wählen Sie im Dropdownmenü die Option Aus Global entfernen aus.



4. Es wird eine Aufforderung angezeigt, in der Sie aufgefordert werden, zu bestätigen, dass Sie den sekundären Cluster vom globalen Cluster trennen möchten. Wählen Sie Entfernen und heraufstufen, um den Cluster aus dem globalen Cluster zu entfernen.



Jetzt dient dieser Cluster nicht mehr als sekundärer Cluster und ist nicht mehr mit dem primären Cluster synchronisiert. Es handelt sich um einen eigenständigen Cluster mit voller Lese-/Schreibfähigkeit.

Nachdem Sie alle sekundären Cluster entfernt oder gelöscht haben, können Sie den primären Cluster auf die gleiche Weise entfernen. Sie können den primären Cluster erst trennen oder vom globalen Cluster entfernen, nachdem Sie alle sekundären Cluster entfernt haben. Der globale Cluster verbleibt möglicherweise in der Clusterliste mit null Regionen und AZs. Sie können löschen, wenn Sie diesen globalen Cluster nicht mehr verwenden möchten.

Using the AWS CLI

Um einen Cluster aus einem globalen Cluster zu entfernen, führen Sie den `remove-from-global-cluster` CLI-Befehl mit den folgenden Parametern aus:

- `--global-cluster-identifier`— Der Name (Bezeichner) Ihres globalen Clusters.
- `--db-cluster-identifier`— Der Name jedes Clusters, der aus dem globalen Cluster entfernt werden soll.

In den folgenden Beispielen wird zuerst ein sekundärer Cluster und dann der primäre Cluster aus einem globalen Cluster entfernt.

Für Linux, macOS oder Unix:

```
aws docdb --region secondary_region \
  remove-from-global-cluster \
    --db-cluster-identifier secondary_cluster_ARN \
```

```
    --global-cluster-identifier global_cluster_id

aws docdb --region primary_region \
  remove-from-global-cluster \
    --db-cluster-identifier primary_cluster_ARN \
    --global-cluster-identifier global_cluster_id
```

Wiederholen Sie den `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` Befehl für jede sekundäre Region in Ihrem globalen Cluster.

Für Windows:

```
aws docdb --region secondary_region ^
  remove-from-global-cluster ^
    --db-cluster-identifier secondary_cluster_ARN ^
    --global-cluster-identifier global_cluster_id

aws docdb --region primary_region ^
  remove-from-global-cluster ^
    --db-cluster-identifier primary_cluster_ARN ^
    --global-cluster-identifier global_cluster_id
```

Wiederholen Sie den `remove-from-global-cluster --db-cluster-identifier secondary_cluster_ARN` Befehl für jede sekundäre Region in Ihrem globalen Cluster.

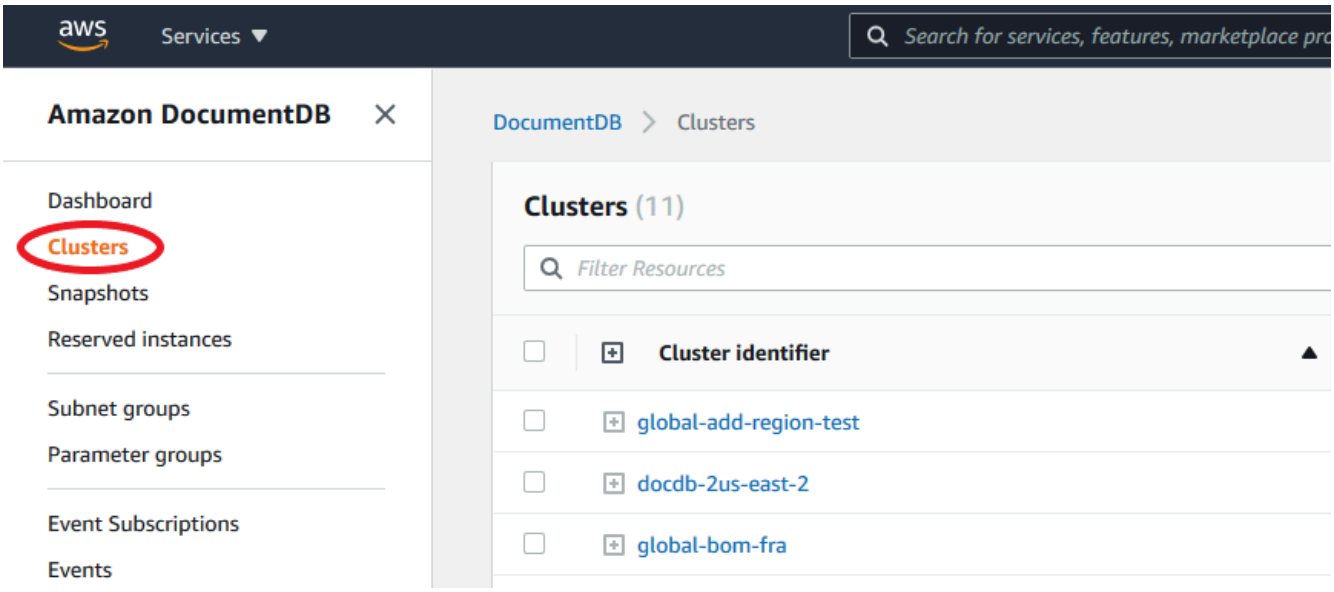
Löschen eines Clusters aus einem globalen Amazon DocumentDB-Cluster

Gehen Sie wie folgt vor, um einen globalen Cluster zu löschen:

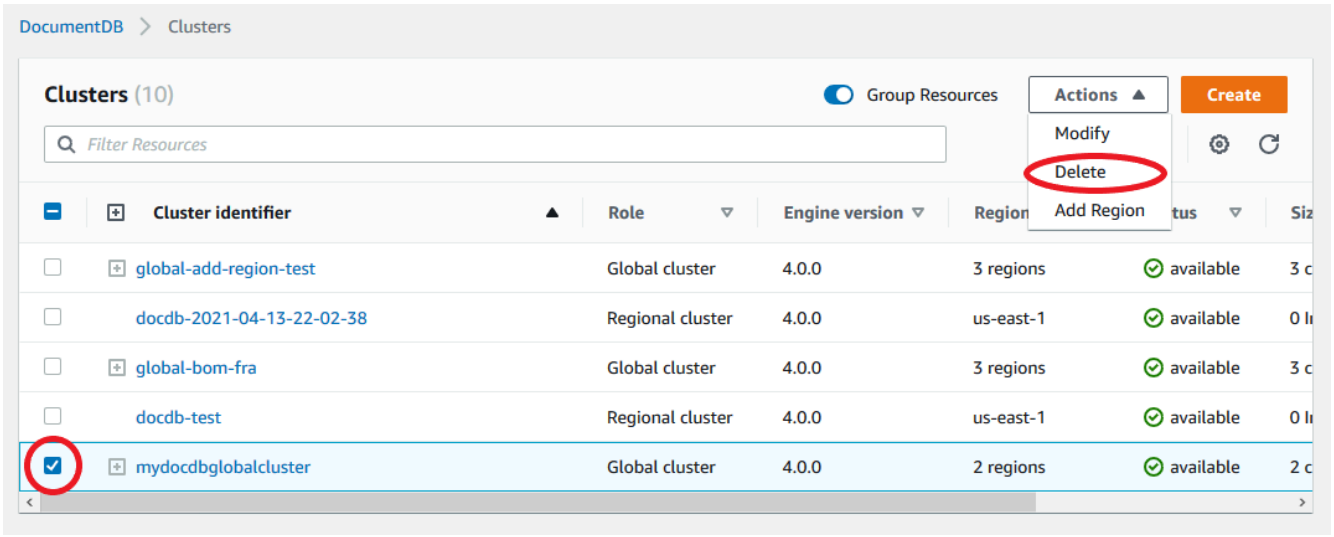
- Entfernen Sie alle sekundären Cluster aus dem globalen Cluster. Jeder Cluster wird zu einem eigenständigen Cluster. Siehe den vorherigen Abschnitt,Einen Cluster aus einem globalen Amazon DocumentDB-Cluster entfernen.
- Löschen Sie aus jedem eigenständigen Cluster alle Replikate.
- Entfernen Sie den primären Cluster aus dem globalen Cluster. Dies wird zu einem eigenständigen Cluster.
- Löschen Sie aus dem primären Cluster zuerst alle Replikate und dann die primäre Instanz. Wenn Sie die primäre Instanz aus dem neuen eigenständigen Cluster löschen, werden in der Regel auch der Cluster und der globale Cluster entfernt.

Using the AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und navigieren Sie zur Amazon DocumentDB DocumentDB-Konsole.
2. Wählen Sie Clusters und suchen Sie den globalen Cluster, den Sie löschen möchten.



3. Wählen Sie Ihren globalen Cluster aus und wählen Sie im Menü Aktionen die Option Löschen aus.



Vergewissern Sie sich, dass alle Cluster aus dem globalen Cluster entfernt wurden. Der globale Cluster sollte keine Regionen AZs und eine Größe von Null Clustern aufweisen. Wenn der globale Cluster Cluster enthält, können Sie ihn noch nicht löschen. Sie müssen zunächst

den Anweisungen im vorherigen Schritt folgen Einen Cluster aus einem globalen Amazon DocumentDB-Cluster entfernen.

Using the AWS CLI

Um einen globalen Cluster zu löschen, führen Sie den `delete-global-cluster` CLI-Befehl mit dem Namen des AWS-Region und der globalen Cluster-ID aus, wie im folgenden Beispiel gezeigt.

Für Linux, macOS oder Unix:

```
aws docdb --region primary_region delete-global-cluster \
   --global-cluster-identifier global_cluster_id
```

Für Windows:

```
aws docdb --region primary_region delete-global-cluster ^
   --global-cluster-identifier global_cluster_id
```

Erstellen eines Headless-Amazon-DocumentDB-Clusters in einer sekundären Region

Obwohl ein globaler Amazon DocumentDB-Cluster mindestens einen sekundären Cluster in einem anderen AWS-Region als dem primären Cluster benötigt, können Sie für den sekundären Cluster eine Headless-Konfiguration verwenden. Ein kopfloser sekundärer Amazon DocumentDB-Cluster ist ein Cluster ohne Instance. Diese Art der Konfiguration kann die Kosten für einen globalen Cluster senken. In einem Amazon DocumentDB-Cluster sind Rechenleistung und Speicher entkoppelt. Ohne die Instance werden Ihnen keine Rechenleistung, sondern nur Speicherplatz in Rechnung gestellt. Wenn es korrekt eingerichtet ist, wird das Speichervolumen eines sekundären Headless-Clusters mit dem des primären Clusters synchronisiert.

Sie fügen den sekundären Cluster wie gewohnt hinzu, wenn Sie einen globalen Amazon DocumentDB-Cluster erstellen. Nachdem der primäre Cluster jedoch mit der Replikation auf den sekundären Cluster begonnen hat, löschen Sie die schreibgeschützte Instance aus dem sekundären Cluster. Dieser sekundäre Cluster gilt nun als „kopflos“, da er keine Instanz mehr hat. Das Speichervolumen wird jedoch mit dem primären Amazon DocumentDB-Cluster synchronisiert.

> **Important**
>
> Wir empfehlen Headless-Cluster nur für Kunden, die regionsweite Ausfälle für mehr als 15 Minuten tolerieren können. Dies liegt daran, dass der Benutzer bei der Wiederherstellung

nach einem regionsweiten Ausfall mit einem sekundären Headless-Cluster nach einem Failover eine neue Instanz erstellen muss. Es kann ~10-15 Minuten dauern, bis eine neue Instanz verfügbar ist.

So fügen Sie Ihrem globalen Cluster einen sekundären Headless-Cluster hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole.
2. Wählen Sie in der linken Navigationsleiste Clusters aus.
3. Wählen Sie den globalen Cluster aus, der einen sekundären Cluster benötigt. Stellen Sie sicher, dass der primäre `Available` Cluster
4. Wählen Sie unter Actions (Aktionen) die Option Add region (Region hinzufügen) aus.
5. Wählen Sie auf der Seite Region hinzufügen die sekundäre Region aus.

   > **Note**
   >
   > Sie können keine Region auswählen, die bereits über einen sekundären Cluster für denselben globalen Cluster verfügt. Außerdem kann es sich nicht um dieselbe Region wie der primäre Cluster handeln.

6. Füllen Sie die verbleibenden Felder für den sekundären Cluster in der neuen Region aus. Dies sind dieselben Konfigurationsoptionen wie für jede Clusterinstanz.
7. Fügen Sie eine Region hinzu. Nachdem Sie die Region zu Ihrem globalen Cluster hinzugefügt haben, wird sie in der Liste von `Clusters` angezeigt AWS Management Console.
8. Überprüfen Sie den Status des sekundären Clusters und seiner Reader-Instance, bevor Sie fortfahren, indem Sie den Befehl AWS Management Console oder den verwenden AWS CLI. Hier ist ein Beispielbefehl, wenn Sie den folgenden Befehl verwenden AWS CLI:

   ```
   $ aws docdb describe-db-clusters --db-cluster-identifier secondary-cluster-id --
   query '*[].[Status]' --output text
   ```

   Es kann mehrere Minuten dauern, bis der Status eines neu hinzugefügten sekundären Clusters von „Wird erstellt“ auf „Verfügbar“ geändert wird. Wenn der Cluster verfügbar ist, können Sie die Reader-Instanz löschen.
9. Wählen Sie die Reader-Instanz im sekundären Cluster aus und klicken Sie dann auf Löschen.

10. Nach dem Löschen der Reader-Instanz bleibt der sekundäre Cluster Teil des globalen Clusters. Ihm sollte keine Instanz zugeordnet sein.

> **Note**
>
> Sie können diesen kopflosen sekundären Amazon DocumentDB-Cluster verwenden, um Ihren globalen Amazon DocumentDB-Cluster manuell nach einem ungeplanten Ausfall in der primären Region wiederherzustellen, falls ein solcher Ausfall auftritt.

Stellen Sie eine Connect zu einem globalen Amazon DocumentDB-Cluster her

Wie Sie eine Verbindung zu einem globalen Cluster herstellen, hängt davon ab, ob Sie in den Cluster schreiben oder aus dem Cluster lesen müssen:

- Für schreibgeschützte Anfragen oder Abfragen stellen Sie eine Verbindung zum Reader-Endpunkt für den Cluster in Ihrem her. AWS-Region
- Stellen Sie zum Ausführen von Data Manipulation Language (DML)- und Data Definition Language (DDL)-Anweisungen eine Verbindung zum Cluster-Endpunkt des primären Clusters her. Dieser Endpunkt befindet sich möglicherweise in einer anderen Anwendung AWS-Region als Ihrer Anwendung.

Wenn Sie sich einen globalen Cluster in der Konsole ansehen, können Sie alle Allzweck-Endpunkte sehen, die mit all seinen Clustern verknüpft sind.

Wie Sie eine Verbindung zu einem globalen Cluster herstellen, hängt davon ab, ob Sie in die Datenbank schreiben oder aus der Datenbank lesen müssen. Für DDL-, DML- und Lesevorgänge, die Sie von der primären Region aus ausführen möchten, sollten Sie eine Verbindung zu Ihrem primären Cluster herstellen. Wir empfehlen, dass Sie eine Verbindung zu Ihrem primären Cluster herstellen, indem Sie den Cluster-Endpunkt im Replikatsatzmodus verwenden, mit der Lesepräferenz von. `secondaryPreferred=true` Dadurch wird der Schreibdatenverkehr an die Writer-Instance Ihres primären Clusters und der Lesetraffic an die Replikat-Instance Ihres primären Clusters weitergeleitet.

Für regionsübergreifenden, schreibgeschützten Datenverkehr sollten Sie eine Verbindung zu einem Ihrer sekundären Cluster herstellen. Wir empfehlen, dass Sie über den Cluster-Endpunkt

im Replikatsatzmodus eine Verbindung zu Ihrem sekundären Cluster herstellen. Da es sich bei allen Instanzen um schreibgeschützte Replikatinstanzen handelt, müssen Sie keine Lesepräferenz angeben. Um die Latenz zu minimieren, wählen Sie den Reader-Endpunkt aus, der sich in Ihrer Region oder in der Region befindet, die Ihnen am nächsten ist.

Überwachung globaler Amazon DocumentDB-Cluster

Amazon DocumentDB (mit MongoDB-Kompatibilität) lässt sich integrieren, CloudWatch sodass Sie Betriebsmetriken für Ihre Cluster sammeln und analysieren können. Sie können diese Metriken mit der CloudWatch Konsole, der Amazon DocumentDB DocumentDB-Konsole, der AWS Command Line Interface (AWS CLI) oder der CloudWatch API überwachen.

Verwenden Sie die folgenden CloudWatch Metriken, um einen globalen Cluster zu überwachen.

Metrik	Beschreibung
`GlobalClusterReplicatedWriteIO`	Die durchschnittliche Anzahl der in Rechnung gestellten I/O-Schreibvorgänge, die vom Cluster-Volume im primären AWS-Region auf das Cluster-Volume in einem sekundäre n Cluster-Volume repliziert wurden AWS-Region, wird in Intervallen von 5 Minuten gemeldet. Die Anzahl der Replikationen in `ReplicatedWriteIOs` jede sekundäre Region entspricht der Anzahl der in der Region `VolumeWriteIOPs` durchgeführten Replikati onen in der primären Region.
`GlobalClusterDataTransferBytes`	Die Datenmenge, die vom primären Cluster AWS-Region zu dem eines sekundären Clusters übertragen wird AWS-Region, gemessen in Byte.
`GlobalClusterReplicationLag`	Das Ausmaß der Verzögerung in Millisekunden bei der Replikation von Änderungsereignissen vom primären Cluster auf das eines sekundäre n Clusters AWS-Region AWS-Region

Weitere Informationen zum Anzeigen dieser Metriken finden Sie unter Daten anzeigen. CloudWatch

Disaster Recovery und globale Amazon DocumentDB-Cluster

Themen

- Durchführen eines verwalteten Failovers für einen globalen Amazon DocumentDB-Cluster
- Durchführen eines manuellen Failovers für einen globalen Amazon DocumentDB-Cluster
- Durchführen eines Switchovers für einen globalen Amazon DocumentDB-Cluster
- Entsperren eines globalen Cluster-Switchovers oder -Failovers

Durch die Verwendung eines globalen Clusters können Sie nach Katastrophen wie regionalen Ausfällen schnell Daten wiederherstellen. Die Wiederherstellung nach einem Notfall wird in der Regel anhand von RTO- und RPO-Werten gemessen.

- Recovery Time Objective (RTO) – Die Zeit, die ein System benötigt, um nach einem Notfall in einen arbeitsfähigen Zustand zurückzukehren. Mit anderen Worten: RTO misst die Ausfallzeit. Für einen globalen Cluster: RTO innerhalb von Minuten.
- Recovery Point Objective (RPO) – Die Datenmenge, die verloren gehen kann (gemessen in Zeit). Für einen globalen Cluster wird RPO in der Regel in Sekunden gemessen.
- Zur Wiederherstellung nach einem ungeplanten Ausfall können Sie ein regionsübergreifendes Failover zu einem der sekundären Cluster in Ihrem globalen Cluster durchführen. Wenn Ihr globaler Cluster über mehrere sekundäre Regionen verfügt, stellen Sie sicher, dass Sie alle sekundären Regionen, die Sie als primäre Regionen heraufstufen möchten, abtrennen. Anschließend stufen Sie eine dieser sekundären Regionen zur neuen Hauptregion herauf. AWS-Region Schließlich erstellen Sie in jeder der anderen sekundären Regionen neue Cluster und fügen diese Cluster Ihrem globalen Cluster hinzu.

Durchführen eines verwalteten Failovers für einen globalen Amazon DocumentDB-Cluster

Dieser Ansatz dient der Geschäftskontinuität im Falle einer echten regionalen Katastrophe oder eines kompletten Service-Level-Ausfalls.

Während eines verwalteten Failovers wird Ihr primärer Cluster auf die sekundäre Region Ihrer Wahl umgestellt, während die bestehende Replikationstopologie Ihres globalen Amazon DocumentDB-

Clusters beibehalten wird. Der gewählte sekundäre Cluster stuft einen seiner schreibgeschützten Knoten auf vollen Writer-Status herauf. In diesem Schritt kann der Cluster die Rolle des primären Clusters übernehmen. Ihre Datenbank ist für kurze Zeit nicht verfügbar, während dieser Cluster seine neue Rolle annimmt. Daten, die nicht vom alten primären auf den ausgewählten sekundären Cluster repliziert wurden, fehlen möglicherweise, wenn dieser sekundäre zum neuen primären Cluster wird. Das alte primäre Volume versucht nach besten Kräften, vor der Synchronisierung mit dem neuen primären Volume einen Snapshot zu erstellen, sodass nicht replizierte Daten auf dem Snapshot erhalten bleiben.

> **Note**
>
> Sie können ein verwaltetes regionsübergreifendes Cluster-Failover auf einem globalen Amazon DocumentDB-Cluster nur durchführen, wenn der primäre und alle sekundären Cluster dieselben Engine-Versionen haben. Wenn Ihre Engine-Versionen nicht kompatibel sind, können Sie das Failover manuell durchführen, indem Sie die Schritte unter Durchführen eines manuellen Failovers für einen globalen Amazon DocumentDB-Cluster ausführen. Wenn die Engine-Versionen der Region nicht übereinstimmen, wird der Failover blockiert. Bitte suchen Sie nach ausstehenden Upgrades und wenden Sie sie an, um sicherzustellen, dass die Engine-Versionen aller Regionen übereinstimmen und der globale Cluster-Failover entsperrt ist. Weitere Informationen finden Sie unter Entsperren eines globalen Cluster-Switchovers oder -Failovers.

Um Datenverlust zu minimieren, empfehlen wir Ihnen, vor der Verwendung dieser Funktion Folgendes zu tun:

- Schalten Sie Anwendungen offline, um zu verhindern, dass Schreibvorgänge an den primären Cluster des globalen Amazon DocumentDB-Clusters gesendet werden.
- Überprüfen Sie die Verzögerungszeiten für alle sekundären Amazon DocumentDB-Cluster. Durch die Auswahl der sekundären Region mit der geringsten Replikationsverzögerung kann der Datenverlust in der aktuell ausgefallenen primären Region minimiert werden. Überprüfen Sie die Verzögerungszeiten für alle sekundären Amazon DocumentDB-Cluster im globalen Cluster, indem Sie sich die `GlobalClusterReplicationLag` Metrik in Amazon CloudWatch ansehen. Diese Metriken zeigen Ihnen, wie weit (in Millisekunden) die Replikation auf einen sekundären Cluster auf den primären Cluster zurückliegt.

  Weitere Informationen zu CloudWatch Metriken für Amazon DocumentDB finden Sie unterAmazon DocumentDB-Metriken.

Während eines verwalteten Failovers wird dem ausgewählten sekundären Cluster seine neue Rolle als primärer Cluster zugewiesen. Er erbt jedoch nicht die verschiedenen Konfigurationsoptionen des primären Clusters. Wenn die Konfiguration nicht übereinstimmt, kann dies Leistungsprobleme, Workload-Inkompatibilitäten und anderes anomales Verhalten zur Folge haben. Um solche Probleme zu vermeiden, empfehlen wir Ihnen, die Unterschiede zwischen Ihren globalen Amazon DocumentDB-Clustern für Folgendes zu beheben:

- Konfigurieren Sie bei Bedarf eine Amazon DocumentDB-Cluster-Parametergruppe für den neuen Primärserver — Sie können Ihre Amazon DocumentDB-Cluster-Parametergruppen unabhängig für jeden Cluster in Ihrem globalen Amazon DocumentDB-Cluster konfigurieren. Wenn Sie also einen sekundären Cluster zur Übernahme der primären Rolle heraufstufen, kann es sein, dass die Parametergruppe aus dem sekundären Cluster anders konfiguriert wird als für den primären. Wenn ja, ändern Sie die Parametergruppe des hochgestuften sekundären Clusters so, dass sie den Einstellungen Ihres primären Clusters entspricht. Um zu erfahren wie dies geht, vgl. Amazon DocumentDB-Cluster-Parametergruppen ändern.
- Überwachungstools und -optionen wie CloudWatch Amazon-Ereignisse und -Alarme konfigurieren — Konfigurieren Sie den beworbenen Cluster mit denselben Protokollierungsfunktionen, Alarmen usw., wie es für den globalen Cluster erforderlich ist. Wie bei Parametergruppen wird die Konfiguration für diese Funktionen während des Failover-Prozesses nicht vom primären Cluster übernommen. Einige CloudWatch Metriken, wie z. B. die Verzögerung bei der Replikation, sind nur für sekundäre Regionen verfügbar. Daher ändert ein Failover die Art und Weise, wie diese Metriken angezeigt und Alarme für sie eingerichtet werden, und es können Änderungen an allen vordefinierten Dashboards erforderlich sein. Weitere Informationen zu Amazon DocumentDB-Clustern und deren Überwachung finden Sie unterÜberwachung von Amazon DocumentDB.

In der Regel übernimmt der gewählte sekundäre Cluster innerhalb einer Minute die primäre Rolle. Sobald der Writer-Knoten der neuen primären Region verfügbar ist, können Sie Ihre Anwendungen mit diesem verbinden und Ihre Workloads fortsetzen. Nachdem Amazon DocumentDB den neuen primären Cluster hochgestuft hat, erstellt es automatisch alle zusätzlichen sekundären Regionscluster neu.

Da globale Amazon DocumentDB-Cluster asynchrone Replikation verwenden, kann die Replikationsverzögerung in jeder sekundären Region variieren. Amazon DocumentDB erstellt diese sekundären Regionen neu, sodass sie genau dieselben point-in-time Daten wie der neue Cluster für die primäre Region haben. Die Dauer der vollständigen Neuerstellungsaufgabe kann je nach Größe des Speichervolumens und der Entfernung zwischen den Regionen einige Minuten bis mehrere Stunden dauern. Wenn der Wiederaufbau der Cluster der sekundären Region aus der

neuen primären Region abgeschlossen ist, stehen sie für den Lesezugriff zur Verfügung. Sobald der neue primäre Writer beworben und verfügbar ist, kann der Cluster der neuen primären Region Lese- und Schreibvorgänge für den globalen Amazon DocumentDB-Cluster abwickeln.

Um die ursprüngliche Topologie des globalen Clusters wiederherzustellen, überwacht Amazon DocumentDB die Verfügbarkeit der alten primären Region. Sobald diese Region fehlerfrei und wieder verfügbar ist, fügt Amazon DocumentDB sie automatisch wieder als sekundäre Region zum globalen Cluster hinzu. Bevor das neue Speichervolume in der alten primären Region erstellt wird, versucht Amazon DocumentDB, zum Zeitpunkt des Fehlers einen Snapshot des alten Speichervolumes zu erstellen. Auf diese Weise können Sie alle fehlenden Daten wiederherstellen. Wenn dieser Vorgang erfolgreich ist, platziert Amazon DocumentDB diesen Snapshot mit dem Namen „rds: docdb-unplanned-global-failover - name-of-old-primary -DB-Cluster-timestamp“ im Snapshot-Abschnitt von. AWS Management Console Sie können diesen Snapshot auch in den Informationen sehen, die von der API-Operation zurückgegeben werden. `DescribeDBClusterSnapshots`

> **Note**
>
> Der Snapshot des alten Speichervolumes ist ein System-Snapshot, der der auf dem alten primären Cluster konfigurierten Aufbewahrungsfrist für Backups unterliegt. Um diesen Snapshot außerhalb des Aufbewahrungszeitraums zu speichern, können Sie ihn kopieren, um ihn als manuellen Snapshot zu speichern. Weitere Informationen zum Kopieren von Snapshots, einschließlich der Preise, finden Sie unter Einen Cluster-Snapshot kopieren.

Nachdem die ursprüngliche Topologie wiederhergestellt ist, können Sie Ihr globales Cluster auf die ursprüngliche primäre Region zurückführen, indem Sie einen Switchover-Vorgang durchführen, wenn dies für Ihr Unternehmen und Ihre Arbeitslast am sinnvollsten ist. Eine Schritt-für-Schritt-Anleitung hierzu finden Sie unter Durchführen eines Switchovers für einen globalen Amazon DocumentDB-Cluster.

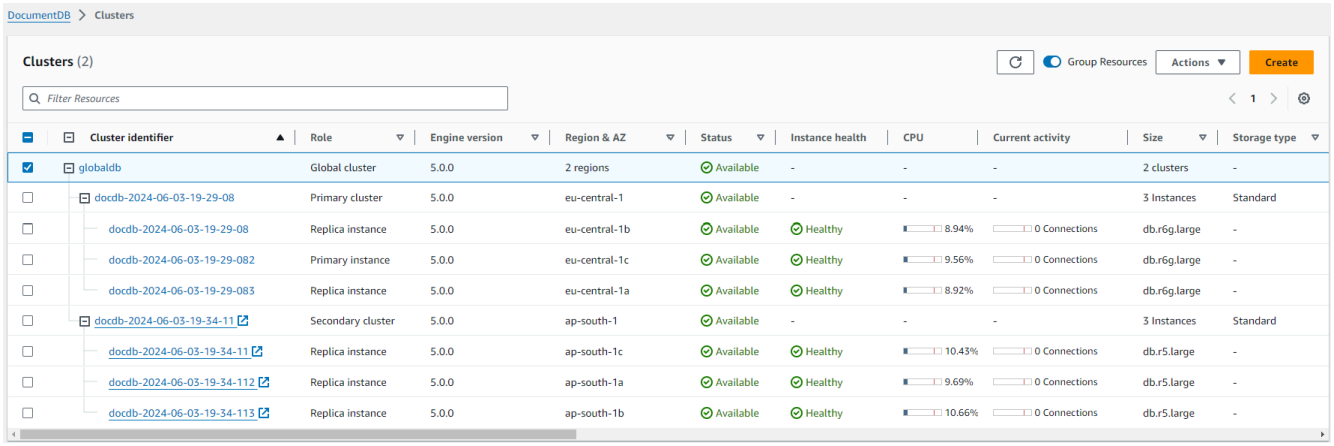
Sie können ein Failover Ihres globalen Amazon DocumentDB-Clusters mithilfe der AWS Management Console, der oder der Amazon DocumentDB DocumentDB-API durchführen. AWS CLI

Using the AWS Management Console

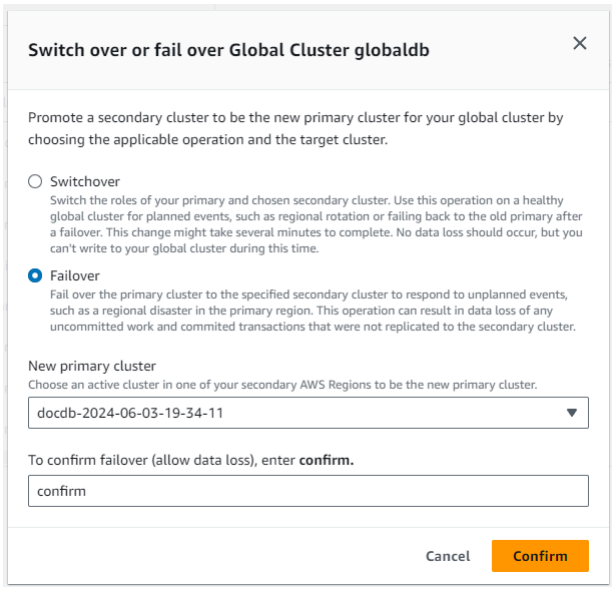
Führen Sie ein verwaltetes Failover auf Ihrem globalen Amazon DocumentDB-Cluster durch

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Klicken Sie im Navigationsbereich auf Cluster.
3. Suchen Sie den globalen Amazon DocumentDB-Cluster, für den Sie ein Failover durchführen möchten, und wählen Sie ihn aus.



4. Wählen Sie im Aktionsmenü Switchover oder Failover aus.
5. Wählen Sie im daraufhin angezeigten Dialogfeld Failover und dann den sekundären Cluster aus der Dropdownliste Neues primäres Clusterfeld aus.

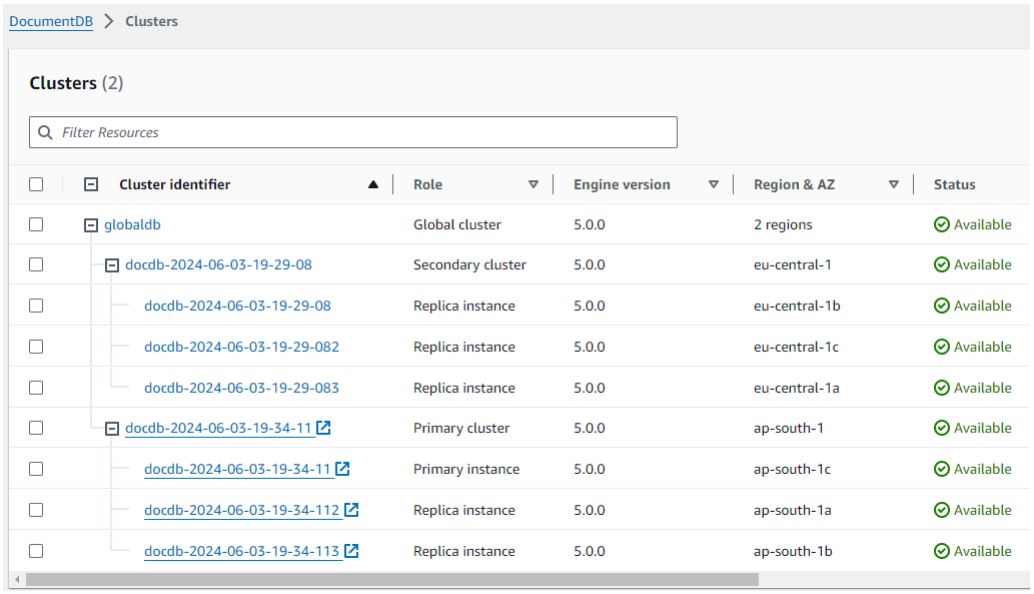


6. Geben Sie „Bestätigen“ in das letzte Feld ein. Wählen Sie dann Confirm (Bestätigen) aus.

   Der Status des primären Clusters ändert sich in "Failing-over“. Dieser Zustand sollte ungefähr eine Minute dauern. Während dieser Zeit zeigt der Status des neuen primären Clusters "Wird geändert... “. Sobald der neue Primärserver hochgestuft wurde, wird dort "Verfügbar" angezeigt und er kann Lese- und Schreibtransaktionen ausführen. In den sekundären Regionen, einschließlich der alten Primärregion, wird "Resyncing... angezeigt ", während es mit dem neuen primären System erneut synchronisiert wird. Ähnlich wie bei der neuen

primären Version kann die Transaktion erst ausgeführt werden, wenn sich der Status auf "Verfügbar" ändert.

7. Wenn der Vorgang abgeschlossen ist, wird der ursprüngliche primäre Cluster zum sekundären Cluster. Der ausgewählte sekundäre Cluster wird zum primären Cluster.



Using the AWS CLI

Führen Sie ein verwaltetes Failover auf Ihrem globalen Amazon DocumentDB-Cluster durch

Führen Sie den `failover-global-cluster`CLI-Befehl aus, um ein Failover für Ihren globalen Amazon DocumentDB-Cluster durchzuführen. Übergeben Sie mit dem Befehl Werte für die folgenden Optionen:

- `--region`
- `--global-cluster-identifier`
- `--target-db-cluster-identifier`
- `--allow-data-loss`

Ersetzen Sie in den folgenden Beispielen jedes *`user input placeholder`* durch die Informationen Ihres Clusters.

Für Linux, macOS oder Unix:

```
aws docdb failover-global-cluster \
```

```
    --region region_of_selected_secondary \
    --global-cluster-identifier global_cluster_id \
    --target-db-cluster-identifier arn_of_secondary_to_promote \
    --allow-data-loss
```

Für Windows:

```
aws docdb failover-global-cluster ^
    --region region_of_selected_secondary ^
    --global-cluster-identifier global_cluster_id ^
    --target-db-cluster-identifier arn_of_secondary_to_promote ^
    --allow-data-loss
```

Durchführen eines manuellen Failovers für einen globalen Amazon DocumentDB-Cluster

Wenn ein ganzer Cluster in einem Cluster nicht AWS-Region mehr verfügbar ist, können Sie einen anderen Cluster im globalen Cluster so heraufstufen, dass er über Lese-/Schreibfunktionen verfügt.

Sie können den globalen Cluster-Failover-Mechanismus manuell aktivieren, wenn ein Cluster in einem anderen Cluster die bessere Wahl als primärer Cluster AWS-Region ist. Sie können beispielsweise die Kapazität eines sekundären Clusters erhöhen und diesen Cluster dann zum primären Cluster hochstufen. Oder das Gleichgewicht zwischen den Aktivitäten AWS-Regionen könnte sich ändern, sodass ein Wechsel des primären Clusters zu einem anderen zu einer geringeren Latenz bei Schreibvorgängen führen AWS-Region kann.

Das folgende Verfahren beschreibt, wie Sie einen der sekundären Cluster in einem globalen Amazon DocumentDB-Cluster bewerben können.

Um einen sekundären Cluster heraufzustufen:

1. Beenden Sie die Ausgabe von DML-Anweisungen und anderen Schreibvorgängen an den primären Cluster während des AWS-Region Ausfalls.
2. Identifizieren Sie einen Cluster aus einem sekundären Cluster AWS-Region , der als neuer primärer Cluster verwendet werden soll. Wenn Sie zwei (oder mehr) sekundäre Cluster AWS-Regionen in Ihrem globalen Cluster haben, wählen Sie den sekundären Cluster mit der geringsten Verzögerungszeit aus.
3. Trennen Sie den ausgewählten sekundären Cluster vom globalen Cluster.

Durch das Entfernen eines sekundären Clusters aus einem globalen Cluster wird die Replikation vom primären auf diesen sekundären Cluster sofort beendet und der Cluster wird zu einem eigenständigen bereitgestellten Cluster mit vollen Lese-/Schreibfunktionen heraufgestuft. Alle anderen sekundären Cluster, die dem primären Cluster in der Region zugeordnet sind, in der der Ausfall aufgetreten ist, sind weiterhin verfügbar und können Anrufe von Ihrer Anwendung annehmen. Sie verbrauchen auch Ressourcen. Da Sie den globalen Cluster neu erstellen, sollten Sie zur Vermeidung von Split-Brain- und anderen Problemen die anderen sekundären Cluster entfernen, bevor Sie den neuen globalen Cluster in den folgenden Schritten erstellen.

Ausführliche Schritte zum Trennen finden Sie unter Einen Cluster aus einem globalen Amazon DocumentDB-Cluster entfernen.

4. Dieser Cluster wird zum primären Cluster eines neuen globalen Clusters, wenn Sie im nächsten Schritt damit beginnen, ihm Regionen hinzuzufügen.
5. Fügen Sie AWS-Region dem Cluster eine hinzu. Wenn Sie dies tun, beginnt der Replikationsprozess vom primären zum sekundären Cluster.
6. Fügen Sie nach AWS-Regionen Bedarf weitere hinzu, um die Topologie neu zu erstellen, die zur Unterstützung Ihrer Anwendung erforderlich ist. Stellen Sie sicher, dass Anwendungsschreibvorgänge vor, während und nach solchen Änderungen an den richtigen Cluster gesendet werden, um Dateninkonsistenzen zwischen den Clustern im globalen Cluster zu vermeiden (Split-Brain-Probleme).
7. Wenn der Ausfall behoben ist und Sie bereit sind, Ihren ursprünglichen Cluster wieder AWS-Region als primären Cluster zuzuweisen, führen Sie dieselben Schritte in umgekehrter Reihenfolge durch.
8. Entfernen Sie einen der sekundären Cluster aus dem globalen Cluster. Dadurch kann er Lese-/Schreibverkehr bereitstellen.
9. Leiten Sie den gesamten Schreibverkehr auf den primären Cluster im Original um. AWS-Region
10. Fügen Sie einen hinzu AWS-Region , um einen oder mehrere sekundäre Cluster AWS-Region wie zuvor einzurichten.

Globale Amazon DocumentDB-Cluster können mithilfe von Amazon DocumentDB verwaltet werden AWS SDKs, sodass Sie Lösungen zur Automatisierung des globalen Cluster-Failover-Prozesses für Anwendungsfälle wie Disaster Recovery und Business Continuity Planning erstellen können. Eine solche Lösung wird unseren Kunden unter der Apache 2.0-Lizenz zur Verfügung gestellt und kann hier in unserem Tool-Repository abgerufen werden. Diese Lösung nutzt Amazon Route 53 für

das Endpunktmanagement und bietet AWS Lambda Funktionen, die auf der Grundlage geeigneter Ereignisse ausgelöst werden können.

Durchführen eines Switchovers für einen globalen Amazon DocumentDB-Cluster

Mithilfe von Switchovers können Sie die Region Ihres primären Clusters routinemäßig ändern. Dieser Ansatz ist für kontrollierte Szenarien gedacht, z. B. für betriebliche Wartung und andere geplante Betriebsverfahren.

Es gibt drei gängige Anwendungsfälle für die Verwendung von Switchovers:

- Für Anforderungen an die „regionale Rotation“, die in bestimmten Branchen gelten. Beispielsweise könnten die Vorschriften für Finanzdienstleistungen vorschreiben, dass Tier-0-Systeme für mehrere Monate in eine andere Region wechseln müssen, um sicherzustellen, dass die Notfallwiederherstellungsverfahren regelmäßig geübt werden.
- Für "follow-the-sun" -Anwendungen mit mehreren Regionen. Beispielsweise möchte ein Unternehmen möglicherweise Schreibvorgänge mit niedrigerer Latenz in verschiedenen Regionen bereitstellen, basierend auf den Geschäftszeiten in verschiedenen Zeitzonen.
- Als zero-data-loss Methode, um nach einem Failover zur ursprünglichen primären Region zurückzukehren.

> **Note**
>
> Switchover sind für die Verwendung in einem gesunden globalen Amazon DocumentDB-Cluster konzipiert. Folgen Sie zur Wiederherstellung nach einem ungeplanten Ausfall dem entsprechenden Verfahren unter Durchführen eines manuellen Failovers für einen globalen Amazon DocumentDB-Cluster.
> Um einen Switchover durchzuführen, müssen in allen sekundären Regionen exakt dieselbe Engine-Version wie in der primären Region ausgeführt werden. Wenn die Engine-Versionen der Region nicht übereinstimmen, wird der Switchover blockiert. Bitte suchen Sie nach ausstehenden Upgrades und wenden Sie sie an, um sicherzustellen, dass die Engine-Versionen aller Regionen übereinstimmen und der globale Cluster-Switchover entsperrt ist. Weitere Informationen finden Sie unter Entsperren eines globalen Cluster-Switchovers oder -Failovers.

Während eines Switchovers wechselt Amazon DocumentDB Ihren primären Cluster in die von Ihnen gewählte sekundäre Region, während die bestehende Replikationstopologie Ihres globalen Clusters beibehalten wird. Bevor der Switchover-Prozess gestartet wird, wartet Amazon DocumentDB, bis alle Cluster der sekundären Region vollständig mit dem Cluster der primären Region synchronisiert sind. Dann wird der DB-Cluster in der primären Region schreibgeschützt, und der ausgewählte sekundäre Cluster stuft einen seiner schreibgeschützten Knoten auf vollen Writer-Status hoch. Durch die Heraufstufung dieses Knotens zu einem Writer kann dieser sekundäre Cluster die Rolle des primären Clusters übernehmen. Da alle sekundären Cluster zu Beginn des Prozesses mit dem primären synchronisiert wurden, setzt der neue primäre Cluster den Betrieb für den globalen Amazon DocumentDB-Cluster fort, ohne Daten zu verlieren. Ihre Datenbank ist für kurze Zeit nicht verfügbar, während der primäre und der ausgewählte sekundäre Cluster ihre neuen Rollen übernehmen.

Um die Anwendungsverfügbarkeit zu optimieren, empfehlen wir Ihnen, vor der Verwendung dieser Funktion Folgendes zu tun:

- Führen Sie diesen Vorgang außerhalb der Spitzenzeiten oder zu einem anderen Zeitpunkt durch, zu dem die Schreibvorgänge auf den primären Cluster minimal sind.
- Schalten Sie Anwendungen offline, um zu verhindern, dass Schreibvorgänge an den primären Cluster des globalen Amazon DocumentDB-Clusters gesendet werden.
- Überprüfen Sie die Verzögerungszeiten für alle sekundären Amazon DocumentDB-Cluster im globalen Cluster, indem Sie sich die `GlobalClusterReplicationLag` Metrik in Amazon CloudWatch ansehen. Diese Metrik zeigt Ihnen, wie weit (in Millisekunden) die Replikation auf einen sekundären Cluster auf den primären Cluster zurückliegt. Dieser Wert ist direkt proportional zu der Zeit, die Amazon DocumentDB benötigt, um den Switchover abzuschließen. Je größer der Verzögerungswert, desto länger dauert die Umstellung.

  Weitere Informationen zu CloudWatch Metriken für Amazon DocumentDB finden Sie unterAmazon DocumentDB-Metriken.

Während einer Umstellung wird der ausgewählte sekundäre DB-Cluster in seine neue Rolle als primärer Cluster hochgestuft. Er übernimmt jedoch nicht die verschiedenen Konfigurationsoptionen des primären DB-Clusters. Wenn die Konfiguration nicht übereinstimmt, kann dies Leistungsprobleme, Workload-Inkompatibilitäten und anderes anomales Verhalten zur Folge haben. Um solche Probleme zu vermeiden, empfehlen wir Ihnen, die Unterschiede zwischen Ihren globalen Amazon DocumentDB-Clustern für Folgendes zu beheben:

- Konfigurieren Sie die Amazon DocumentDB-DB-Cluster-Parametergruppe für den neuen Primärserver, falls erforderlich — Sie können Ihre Amazon DocumentDB-Cluster-Parametergruppen unabhängig für jeden Cluster in Ihrem globalen Amazon DocumentDB-Cluster konfigurieren. Wenn Sie also einen sekundären DB-Cluster zur Übernahme der primären Rolle hochstufen, kann die Parametergruppe des sekundären Clusters im Vergleich zum primären Cluster möglicherweise anders konfiguriert werden. Ist dies der Fall, ändern Sie die Parametergruppe des hochgestuften sekundären DB-Clusters so, dass sie den Einstellungen des primären Clusters entspricht. Um zu erfahren wie, siehe Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.
- Überwachungstools und -optionen wie Amazon CloudWatch Events und Alarme konfigurieren — Konfigurieren Sie den beworbenen Cluster mit den gleichen Protokollierungsfunktionen, Alarmen usw., wie es für den globalen Cluster erforderlich ist. Wie bei Parametergruppen wird die Konfiguration für diese Funktionen während der Umstellung nicht vom primären Cluster übernommen. Einige CloudWatch Metriken, wie z. B. die Verzögerung bei der Replikation, sind nur für primäre Regionen verfügbar. Daher ändert sich bei einer Umstellung die Art und Weise, wie diese Metriken angezeigt und Alarme für sie eingerichtet werden, und es können Änderungen an allen vordefinierten Dashboards erforderlich sein. Weitere Informationen finden Sie unter Überwachung von Amazon DocumentDB.

> **Note**
>
> In der Regel kann der Rollenwechsel bis zu mehreren Minuten dauern.

Wenn der Switchover-Prozess abgeschlossen ist, kann der beworbene Amazon DocumentDB-Cluster Schreibvorgänge für den globalen Cluster verarbeiten.

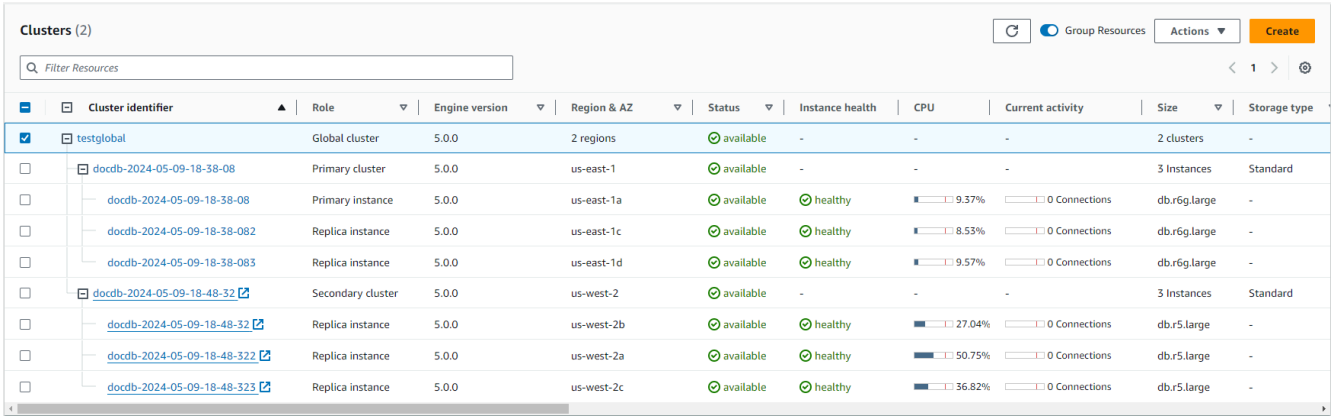
Sie können Ihren globalen Amazon DocumentDB-Cluster mit dem AWS Management Console oder dem AWS CLI folgenden Befehl umschalten:

Using the AWS Management Console

Führen Sie einen Switchover auf Ihrem globalen Amazon DocumentDB-Cluster durch

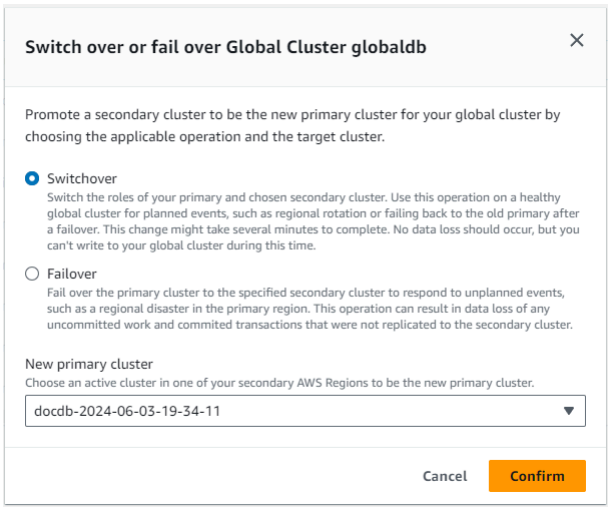
1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

3. Suchen Sie den globalen Amazon DocumentDB-Cluster, zu dem Sie wechseln möchten, und wählen Sie ihn aus.



4. Wählen Sie im Aktionsmenü Switchover oder Failover aus.

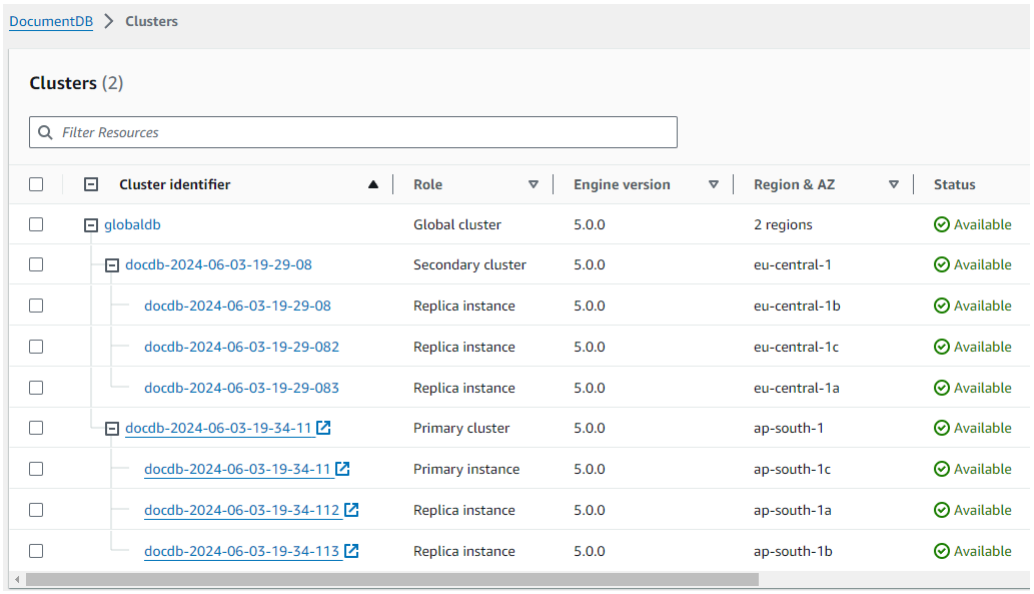
5. Wählen Sie im daraufhin angezeigten Dialogfeld Switchover und dann den sekundären Cluster aus der Dropdownliste Neues primäres Clusterfeld aus.



6. Wählen Sie Bestätigen aus.

   Der Status des primären Clusters ändert sich in "Switching-over“. Dieser Zustand sollte ungefähr drei Minuten dauern. Während dieser Zeit zeigt der Status aller regionalen Cluster "Wird geändert... “. Sobald die Regionen synchronisiert sind und die neue Primärregion beworben wurde, wird in allen Statusfeldern „Verfügbar“ angezeigt und Transaktionen können bearbeitet werden.

7. Wenn der Vorgang abgeschlossen ist, wird der ursprüngliche primäre Cluster zum sekundären Cluster. Der ausgewählte sekundäre Cluster wird zum primären Cluster.



Using the AWS CLI

Führen Sie einen Switchover auf Ihrem globalen Amazon DocumentDB-Cluster durch

Führen Sie den `switchover-global-cluster`CLI-Befehl aus, um Ihren globalen Amazon DocumentDB-Cluster umzuschalten. Übergeben Sie mit dem Befehl Werte für die folgenden Optionen:

- `--region`
- `--global-cluster-identifier`
- `--target-db-cluster-identifier`

Ersetzen Sie in den folgenden Beispielen jedes *`user input placeholder`* durch die Informationen Ihres Clusters.

Für Linux, macOS oder Unix:

```
aws docdb switchover-global-cluster \
   --region region_of_primary \
   --global-cluster-identifier global_cluster_id \
   --target-db-cluster-identifier arn_of_secondary_to_promote
```

Für Windows:

```
aws docdb switchover-global-cluster ^
   --region region_of_primary ^
   --global-cluster-identifier global_cluster_id ^
   --target-db-cluster-identifier arn_of_secondary_to_promote
```

Entsperren eines globalen Cluster-Switchovers oder -Failovers

Globale Cluster-Switchover und -Failover werden blockiert, wenn sich nicht alle regionalen Cluster im globalen Cluster auf derselben Engine-Version befinden. Wenn die Versionen nicht übereinstimmen, wird beim Aufrufen eines Switchovers oder Failovers möglicherweise dieser Fehler angezeigt: Auf dem angegebenen Ziel-DB-Cluster wird eine Engine-Version mit einem anderen Patch-Level ausgeführt als auf dem Quell-DB-Cluster. Wir empfehlen, routinemäßig die neuesten Engine-Versionen zu installieren, um sicherzustellen, dass Sie die neuesten Updates ausführen, damit Ihre globalen Cluster in einem fehlerfreien Zustand bleiben.

Um diesen Fehler zu beheben, aktualisieren Sie bitte zuerst alle sekundären Regionen und dann die primäre Region auf dieselbe Engine-Version, indem Sie alle ausstehenden Wartungsmaßnahmen anwenden. Führen Sie die Anweisungen auf einer der folgenden Registerkarten aus, um ausstehende Wartungsmaßnahmen anzuzeigen und die zur Behebung des Problems erforderlichen Änderungen vorzunehmen:

Using the AWS Management Console

Um die Blockierung eines globalen Cluster-Switchovers oder -Failovers aufzuheben, müssen Sie feststellen, ob noch Wartungsaktionen für Ihre Cluster ausstehen, und diese anwenden. Gehen Sie wie folgt vor, um Wartungsaktionen anzuzeigen und anzuwenden:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Suchen Sie in der Cluster-Tabelle in der Spalte Cluster-ID nach Ihrem globalen Cluster. Notieren Sie sich unter Ihrem globalen Cluster jeden sekundären Cluster und den primären Cluster für den jeweiligen globalen Cluster und führen Sie für jeden Cluster die folgenden Schritte aus.
4. Für jeden sekundären Cluster:

    a. Wenn ein Update für Ihren Cluster verfügbar ist, wird es in der Spalte Wartung als Verfügbar, Erforderlich oder Nächstes Fenster angezeigt.

b. Um eine Aktion auszuführen, wählen Sie den Cluster aus, dessen Details angezeigt werden sollen, und wählen Sie dann Wartung und Backups aus. Die Einträge „Ausstehende Wartung“ werden angezeigt.

c. Wenn unter Beschreibung angezeigt wird, dass ein „Neues Wartungsupdate verfügbar ist“, wählen Sie es aus und klicken Sie dann auf Jetzt anwenden.

5. Für Ihren primären Cluster:

a. Wenn ein Update für Ihren Cluster verfügbar ist, wird es in der Spalte Wartung als Verfügbar, Erforderlich oder Nächstes Fenster angezeigt.

b. Um eine Aktion auszuführen, wählen Sie den Cluster aus, dessen Details angezeigt werden sollen, und wählen Sie dann Wartung und Backups aus. Die Einträge „Ausstehende Wartung“ werden angezeigt.

c. Wenn unter Beschreibung angezeigt wird, dass ein „Neues Wartungsupdate verfügbar ist“, wählen Sie es aus und klicken Sie dann auf Jetzt anwenden.

Using the AWS CLI

Um die Blockierung eines globalen Cluster-Switchovers oder -Failovers aufzuheben, müssen Sie feststellen, ob noch Wartungsaktionen für den Cluster ausstehen, und diese anwenden. Gehen Sie wie folgt vor, um Wartungsaktionen zunächst auf den sekundären Clustern und dann auf dem primären Cluster für Ihren globalen Cluster anzuzeigen und anzuwenden:

1. Führen Sie Folgendes zuerst für den regionalen Cluster jeder sekundären Region und dann für den regionalen Cluster der primären Regionen aus.
2. Führen Sie den `describe-pending-maintenance-actions`CLI-Befehl mit der `--resource-identifier` Option aus, um festzustellen, ob Wartungsaktionen für Ihren regionalen Amazon DocumentDB-Cluster verfügbar sind.

   Ersetzen Sie in den folgenden Beispielen jedes *`user input placeholder`* durch die Informationen Ihres Clusters.

   Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-action \
   --resource-identifier arn:aws:rds:us-
east-1:001234567890:cluster:docdb-2025-03-27-19-21-15
```

Für Windows:

```
aws docdb describe-pending-maintenance-action ^
   --resource-identifier arn:aws:rds:us-
east-1:001234567890:cluster:docdb-2025-03-27-19-21-15
```

Das Ergebnis sieht in etwa so aus:

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:001234567890:cluster:docdb-2025-03-27-19-21-15",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "system-update",
                    "CurrentApplyDate": "2025-04-11T03:01:00Z",
                    "Description": "db-version-upgrade",
                    "ForcedApplyDate": "2025-06-18T03:01:00Z",
                    "AutoAppliedAfterDate": "2025-05-11T03:01:00Z"
                    "OptInStatus": "pending"
                }
            ]
        }
    ]
}
```

3. Wenn eine Wartungsaktion erforderlich ist, führen Sie den `apply-pending-maintenance-action`CLI-Befehl mit den folgenden Optionen aus:

   - `--resource-identifier`
   - `--apply-action`
   - `--opt-in-type`
   - `--region`

   Ersetzen Sie in den folgenden Beispielen jedes Beispiel *`user input placeholder`* durch die Informationen Ihres Clusters.

   Für Linux, macOS oder Unix:

```
aws docdb apply-pending-maintenance-action \
   --resource-identifier arn:aws:rds:us-
east-1:001234567890:cluster:docdb-2025-03-27-19-21-15 \
   --apply-action system-update \
   --opt-in-type immediate \
   --region us-east-1
```

Für Windows:

```
aws docdb apply-pending-maintenance-action ^
   --resource-identifier arn:aws:rds:us-
east-1:001234567890:cluster:docdb-2025-03-27-19-21-15 ^
   --apply-action system-update ^
   --opt-in-type immediate ^
   --region us-east-1
```

4. Sobald die Wartungsaktion abgeschlossen ist, führen Sie den describe-pending-maintenance-actionsBefehl erneut aus, um sicherzustellen, dass keine weiteren Aktionen für Ihren Cluster ausstehen.

   Das gewünschte Ergebnis ist:

```
{
    "PendingMaintenanceActions": []
}
```

Using the Amazon DocumentDB API

Um die Blockierung eines globalen Cluster-Switchovers oder -Failovers aufzuheben, müssen Sie feststellen, ob noch Wartungsaktionen für den Cluster ausstehen, und diese anwenden. Gehen Sie wie folgt vor APIs , um Wartungsaktionen anzuzeigen und anzuwenden:

1. Führen Sie den folgenden Befehl zuerst für den regionalen Cluster jeder sekundären Region und dann für den regionalen Cluster der primären Regionen aus.
2. Rufen Sie die PendingMaintenanceActionAPI auf, um festzustellen, ob Wartungsaktionen für Ihren globalen Amazon DocumentDB-Cluster verfügbar sind.
3. Wenden Sie alle Änderungen an, indem Sie die ApplyPendingMaintenanceActionAPI aufrufen.

Verwaltung von Amazon DocumentDB-Clustern

Um einen Amazon DocumentDB-Cluster zu verwalten, benötigen Sie eine IAM-Richtlinie mit den entsprechenden Berechtigungen für die Amazon DocumentDB-Steuerebene. Diese Berechtigungen ermöglichen das Erstellen, Ändern und Löschen von Clustern und Instances. Die `AmazonDocDBFullAccess` Richtlinie bietet alle erforderlichen Berechtigungen für die Verwaltung eines Amazon DocumentDB-Clusters.

Die folgenden Themen zeigen, wie Sie bei der Arbeit mit Amazon DocumentDB-Clustern verschiedene Aufgaben ausführen, darunter das Erstellen, Löschen, Ändern, Herstellen einer Verbindung zu und das Anzeigen von Clustern.

Themen

- Cluster verstehen
- Amazon DocumentDB-Cluster-Einstellungen
- Amazon DocumentDB-Cluster-Speicherkonfigurationen
- Den Status eines Clusters ermitteln
- Amazon DocumentDB-Cluster-Lebenszyklus
- Skalierung von Amazon DocumentDB-Clustern
- Klonen eines Volumes für einen Amazon DocumentDB-Cluster
- Grundlegendes zur Amazon DocumentDB-Cluster-Fehlertoleranz

Cluster verstehen

Amazon DocumentDB trennt Rechenleistung und Speicher und verlagert Datenreplikation und Backup auf das Cluster-Volume. Ein Cluster-Volume bietet eine dauerhafte, zuverlässige und hochverfügbare Speicherschicht, die Daten auf sechs Arten über drei Availability Zones repliziert. Replikate ermöglichen eine höhere Datenverfügbarkeit und Leseskalierung. Jeder Cluster kann auf bis zu 15 Replikate skaliert werden.

Substantiv	Beschreibung	API-Operationen (Verben)
Cluster	Besteht aus einer oder mehreren Instances und einem Cluster-Storage-Vo	`create-db-cluster` `delete-db-cluster`

Substantiv	Beschreibung	API-Operationen (Verben)
	lume, das die Daten für diese Instances verwaltet.	`describe-db-clusters` `modify-db-cluster`

Instance	Das Lesen und Schreiben von Daten auf das Cluster-Speichervolumen erfolgt über Instances. In einem gegebenen Cluster gibt es zwei Arten von Instances : Primär und Replikat. Ein Cluster hat immer eine primäre Instanz und kann 0—15 Replikate haben.	`create-db-instance`
		`delete-db-instance`
		`describe-db-instances`
		`modify-db-instance`
		`describe-orderable-db-instance-options`
		`reboot-db-instance`

Cluster-Volumen	Ein virtuelles Datenbank speichervolumen, das sich über drei Availability Zones erstreckt, wobei jede Availability Zone zwei Kopien der Cluster-Daten aufweist.	N/A
Primäre Instance	Unterstützt Lese- und Schreiboperationen und führt alle Datenänderungen im Cluster-Volume durch. Jeder Cluster verfügt über eine primäre Instance.	N/A

Substantiv	Beschreibung	API-Operationen (Verben)
Replikat-Instance	Unterstützt nur Lesevorgänge. Jeder Amazon DocumentDB-Cluster kann zusätzlich zur primären Instance bis zu 15 Replikat-Instances haben. Mehrere Replikate verteilen den Lese-Workload. Durch die Platzierung von Replikaten in separaten Availability Zones können Sie außerdem die Verfügbarkeit der Datenbank erhöhen.	N/A
Cluster-Endpunkt	Ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zur aktuellen primären Instance für den Cluster herstellt. Jeder Amazon DocumentDB-Cluster hat einen Cluster-Endpunkt und eine primäre Instance.	N/A
Leser-Endpunkt	Ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zu einem der verfügbaren Replikate für diesen Cluster herstellt. Jeder Amazon DocumentDB-Cluster hat einen Leser-Endpunkt. Wenn es mehr als ein Replikat gibt, leitet der Leser-Endpunkt jede Verbindungsanfrage an eines der Amazon DocumentDB DocumentDB-Replikate weiter.	N/A

Substantiv	Beschreibung	API-Operationen (Verben)
Instance-Endpunkt	Ein Endpunkt für eine Instance in einem Amazon DocumentDB-Cluster, der eine Verbindung zu einer bestimmten Instance herstellt. Jede Instance in einem Cluster hat unabhängig vom Instance-Typ einen eigenen eindeutigen Instance-Endpunkt.	N/A

Amazon DocumentDB-Cluster-Einstellungen

Beim Erstellen oder Ändern eines Clusters müssen Sie wissen, welche Parameter unveränderlich und welche nach dem Anlegen des Clusters veränderbar sind. Die folgende Tabelle listet alle Einstellungen bzw. Parameter auf, die für einen Cluster spezifisch sind. Wie in der Tabelle angegeben, sind nur einige Parameter änderbar.

> **Note**
>
> Diese Einstellungen sollten nicht mit Amazon DocumentDB-Cluster-Parametergruppen und ihren Parametern verwechselt werden. Weitere Informationen zu Cluster-Parametergruppen finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.

Parameter	Anpassbar	Hinweise
`DBClusterIdentifier`	Ja	Benennungseinschränkungen: • Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Parameter	Anpassbar	Hinweise
		• Muss für alle Cluster in Amazon RDS, Amazon Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.
Engine	Nein	Der Wert muss `docdb` sein.
BackupRetentionPeriod	Ja	Muss zwischen 1 und 35 Tagen liegen.
DBClusterParameterGroupName	Ja	Benennungseinschränkungen: • Die Länge beträgt [1—255] alphanumerische Zeichen. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
DBSubnetGroupName	Nein	Sie können das Subnetz des Clusters nach dem Anlegen eines Clusters nicht mehr ändern.
EngineVersion	Nein	Der Wert kann `5.0.0` (Standard)`4.0.0`, oder sein. `3.6.0`
KmsKeyId	Nein	Wenn Sie Ihren Cluster verschlüsseln möchten, können Sie den AWS KMS Schlüssel, den Sie zur Verschlüsselung Ihres Clusters verwendet haben, nicht ändern.

Parameter	Anpassbar	Hinweise
`MasterUsername`	Nein	Sie können den `MasterUsername` nach dem Anlegen eines Clusters nicht mehr ändern. Benennungseinschränkungen: • Die Länge beträgt [1—63] alphanumerische Zeichen. • Muss mit einem Buchstaben beginnen. • Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

`MasterUserPassword`	Ja	Einschränkungen:
		
		• Die Länge beträgt [8—100] druckbare ASCII-Zeichen.
		• Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
		  • **`/`** (Schrägstrich)
		  • **`"`** (doppeltes Anführungszeichen)
		  • **`@`** ('At'-Symbol)

`Port`	Ja	Die Portnummer gilt für alle Instances im Cluster.
`PreferredBackupWindow`	Ja	
`PreferredMaintenanceWindow`	Ja	
`StorageEncrypted`	Nein	Wenn Sie Ihren Cluster verschlüsseln, kann er nicht unverschlüsselt sein.

Parameter	Anpassbar	Hinweise
`StorageType`	Ja	Der Speichertyp für den DB-Cluster: Standard (`standard`) oder I/O-Optimiert (). `iopt1` Standard: `standard` Dieser Parameter kann mit `CreateDBC luster` und konfiguriert werden. `ModifyDBC luster` Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Speicherkonfigurationen.
`Tags`	Ja	
`VpcSecurityGroupIds`	Nein	Nachdem ein Cluster erstellt wurde, können Sie die VPC nicht ändern, in der sich der Cluster befindet.

Amazon DocumentDB-Cluster-Speicherkonfigurationen

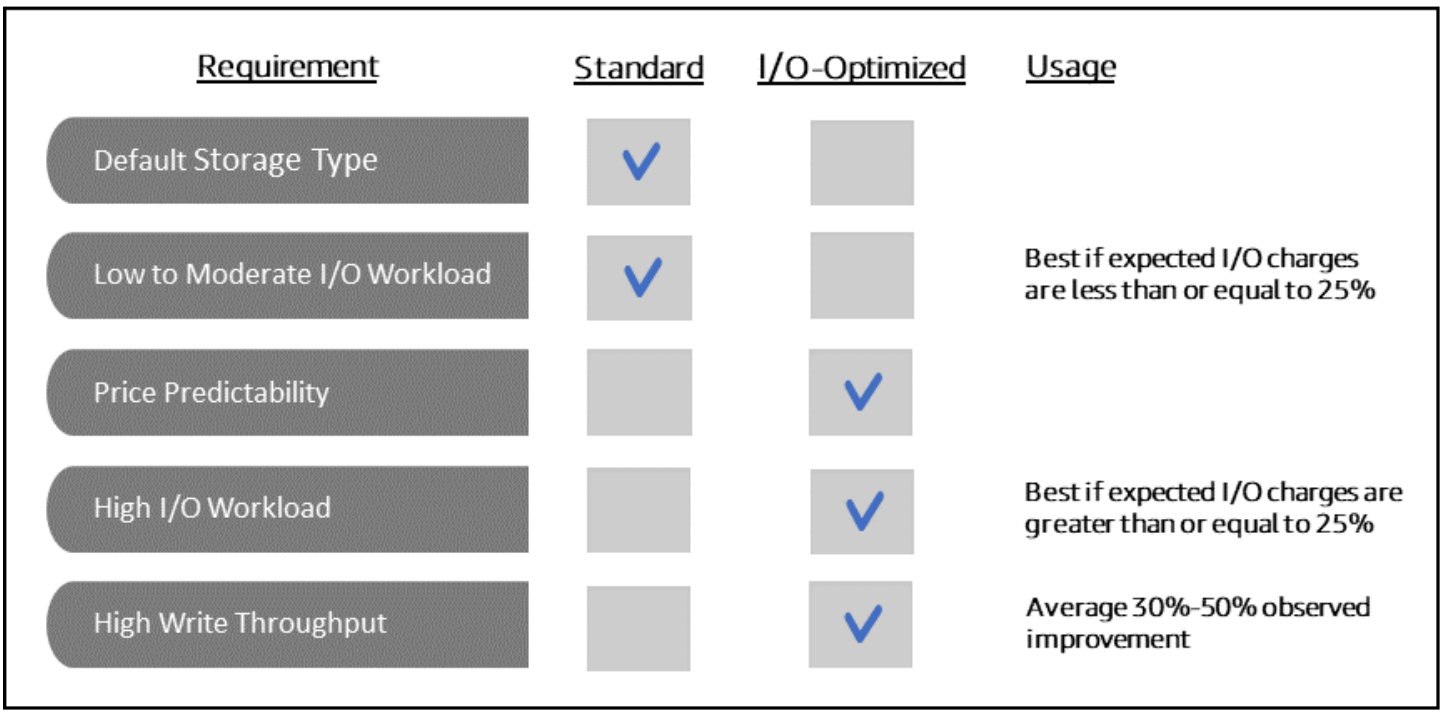
Ab Amazon DocumentDB 5.0 unterstützen instanzbasierte Cluster zwei Speicherkonfigurationstypen:

- Amazon DocumentDB DocumentDB-Standardspeicher: Konzipiert für Kunden mit geringem bis moderatem I/O Verbrauch. Wenn Sie davon ausgehen, dass Ihre I/O Kosten weniger als 25% Ihres gesamten Amazon DocumentDB-Clusters ausmachen werden, ist diese Wahl möglicherweise ideal für Sie. Bei der Standardspeicherkonfiguration von Amazon DocumentDB werden Ihnen zusätzlich zu den Instance- und Speichergebühren zusätzliche Gebühren in Rechnung gestellt. pay-per-request I/O Das bedeutet, dass Ihre Abrechnung je nach Nutzung von einem Zyklus zum anderen variieren kann. Die Konfiguration ist auf die schwankenden I/O Anforderungen Ihrer Anwendung zugeschnitten.
- I/O-optimierter Speicher von Amazon DocumentDB: Konzipiert für Kunden, die Wert auf Preisvorhersehbarkeit legen oder umfangreiche Anwendungen haben. I/O Der I/O-optimized configuration offers improved performance, increased throughput, and reduced latency for customers with I/O intensive workloads. If you expect your I/O costs to exceed 25% of your total Amazon DocumentDB cluster costs, this option offers enhanced price performance. With the Amazon DocumentDB I/O-optimized storage configuration, you won't be charged based on I/O

Betrieb gewährleistet vorhersehbare Kosten in jedem Abrechnungszeitraum. Die Konfiguration stabilisiert die Kosten und verbessert gleichzeitig die Leistung.

Sie können Ihre vorhandenen Datenbank-Cluster einmal alle 30 Tage auf Amazon DocumentDB umstellen. I/O-optimized storage. You can switch back to Amazon DocumentDB standard storage at any time. The next date to modify the storage configuration to I/O Optimiert kann mit dem `describe-db-clusters` Befehl über AWS CLI oder über die auf der Konfigurationsseite des Clusters nachverfolgt werden. AWS Management Console

Sie können einen neuen Datenbank-Cluster einschließlich der I/O-optimierten Amazon DocumentDB DocumentDB-Konfiguration erstellen oder Ihre vorhandenen Datenbank-Cluster mit ein paar Klicks AWS Management Console, einer einzigen Parameteränderung in AWS Command Line Interface (AWS CLI) oder so konvertieren. AWS SDKs Während oder nach der Änderung der Speicherkonfiguration sind keine Ausfallzeiten oder Neustarts von Instances erforderlich.

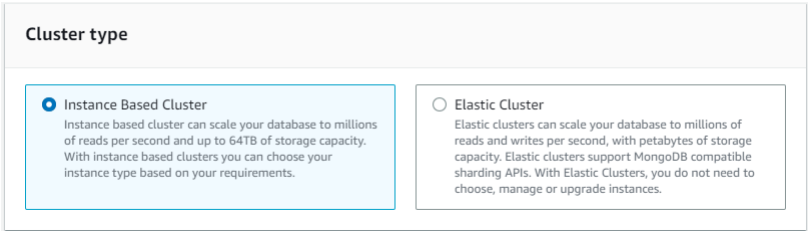


Einen I/O-optimierten Cluster erstellen

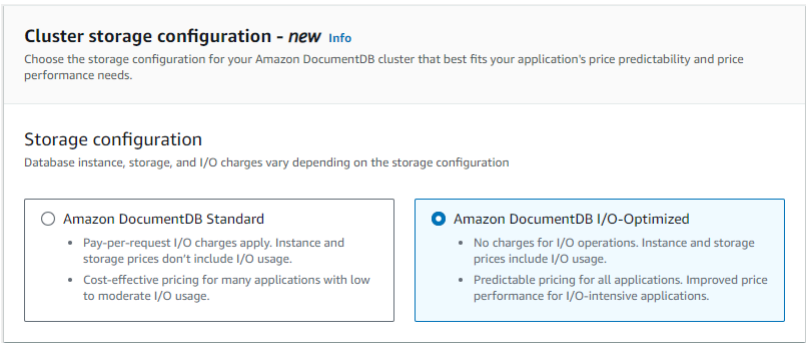
Using the AWS Management Console

Um einen I/O-optimierten Cluster zu erstellen oder zu ändern, verwenden Sie: AWS Management Console

1. Wählen Sie in der Amazon DocumentDB-Managementkonsole unter Clusters entweder Create oder wählen Sie den Cluster aus und wählen Sie Actions und anschließend Modify aus.
2. Wenn Sie einen neuen Cluster erstellen, stellen Sie sicher, dass Sie im Abschnitt Clustertyp die Option Instance Based Cluster auswählen (dies ist die Standardoption).



3. Wählen Sie im Abschnitt Konfiguration unter Cluster-Speicherkonfiguration die Option Amazon DocumentDB I/O-Optimized aus.



4. Schließen Sie die Erstellung oder Änderung Ihres Clusters ab und wählen Sie Cluster erstellen oder Cluster ändern aus.

   Den vollständigen Vorgang zum Erstellen eines Clusters finden Sie unterErstellen eines Clusters und einer primären Instance mit dem AWS Management Console.

   Den vollständigen Vorgang zum Ändern des Clusters finden Sie unterÄndern eines Amazon DocumentDB-Clusters.

Using the AWS CLI

Um einen I/O-optimierten Cluster zu erstellen, verwenden Sie: AWS CLI

Ersetzen Sie in den folgenden Beispielen jedes *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 5.0.0 \
      --storage-type iopt1 \
      --deletion-protection \
      --master-username username \
      --master-user-password password
```

Für Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
      --engine-version 5.0.0 ^
      --storage-type iopt1 ^
      --deletion-protection ^
      --master-username username ^
      --master-user-password password
```

Kostenanalyse zur Bestimmung der Speicherkonfiguration

Mit Amazon DocumentDB haben Sie die Flexibilität, Ihre Speicherkonfiguration für jeden vorhandenen Datenbank-Cluster auszuwählen. Um Ihre Cluster ordnungsgemäß zwischen Standard- und I/O-optimierten Clustern zuzuordnen, können Sie Ihre Amazon DocumentDB DocumentDB-Kosten clusterweise verfolgen. Zu diesem Zweck können Sie vorhandenen Clustern Tags hinzufügen, die Kostenzuordnungskennzeichnung in Ihrem AWS Fakturierung und Kostenmanagement Dashboard aktivieren und Ihre Kosten für einen bestimmten Cluster in der analysieren. AWS Cost Explorer Service Informationen zur Kostenanalyse finden Sie in unserem Blog Verwenden von Kostenzuordnungs-Tags.

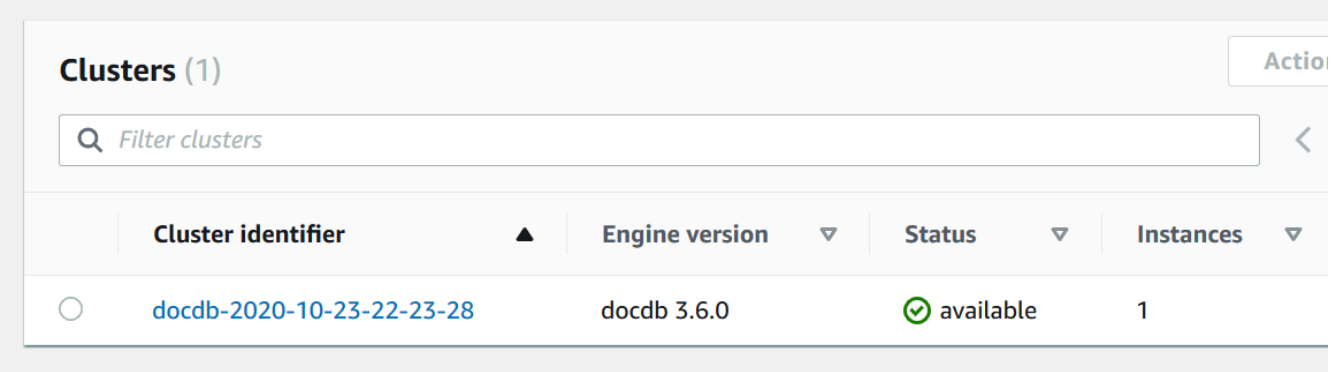
Den Status eines Clusters ermitteln

Sie können den Status eines Clusters mit dem AWS Management Console oder ermitteln AWS CLI.

Using the AWS Management Console

Gehen Sie wie folgt vor, um den Status Ihres Amazon DocumentDB-Clusters mithilfe der AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Suchen Sie in der Spalte Cluster identifier (Cluster-ID) den Namen des Clusters, der Sie interessiert. Um dann den Status des Clusters zu ermitteln, lesen Sie diese Zeile in die Spalte Status, wie unten gezeigt.



Using the AWS CLI

Verwenden Sie den `describe-db-clusters` Vorgang, um den Status Ihres Amazon DocumentDB-Clusters mithilfe von zu überprüfen. AWS CLI

Der folgende Code ermittelt den Status des Clusters `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

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

```
[
    [
        "sample-cluster",
```

```
        "available"
    ]
]
```

Amazon DocumentDB-Cluster-Lebenszyklus

Der Lebenszyklus eines Amazon DocumentDB-Clusters umfasst das Erstellen, Beschreiben, Ändern und Löschen des Clusters. Dieser Abschnitt enthält Informationen darüber, wie Sie diese Prozesse abschließen können.

Themen

- Einen Amazon DocumentDB-Cluster erstellen
- Beschreibung von Amazon DocumentDB-Clustern
- Ändern eines Amazon DocumentDB-Clusters
- Ermittlung ausstehender Wartungsarbeiten
- Durchführen eines Patch-Updates für die Engine-Version eines Clusters
- Einen Amazon DocumentDB-Cluster stoppen und starten
- Löschen eines Amazon DocumentDB-Clusters

Einen Amazon DocumentDB-Cluster erstellen

Ein Amazon DocumentDB-Cluster besteht aus Instances und einem Cluster-Volume, das die Daten für den Cluster darstellt. Das Cluster-Volumen wird auf sechs Arten über drei Availability Zones als ein einziges, virtuelles Volume repliziert. Der Cluster enthält eine primäre Instance und optional bis zu 15 Replikat-Instances.

In den folgenden Abschnitten wird gezeigt, wie Sie einen Amazon DocumentDB-Cluster mit dem AWS Management Console oder dem AWS CLI erstellen. Sie können dann weitere Replikat-Instances für diesen Cluster hinzufügen. Wenn Sie die Konsole verwenden, um Ihren Amazon DocumentDB-Cluster zu erstellen, wird gleichzeitig automatisch eine primäre Instance für Sie erstellt. Wenn Sie den verwenden, AWS CLI um Ihren Amazon DocumentDB-Cluster zu erstellen, müssen Sie, nachdem der Status des Clusters verfügbar ist, die primäre Instance für diesen Cluster erstellen.

Voraussetzungen

Im Folgenden sind die Voraussetzungen für die Erstellung eines Amazon DocumentDB-Clusters aufgeführt.

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

Um sich für eine anzumelden AWS-Konto

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

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

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

VPC-Voraussetzungen

Sie können einen Amazon DocumentDB-Cluster nur in einer Amazon Virtual Private Cloud (Amazon VPC) erstellen. Ihre Amazon VPC muss mindestens ein Subnetz in jeder von mindestens zwei Availability Zones haben, damit Sie sie mit einem Amazon DocumentDB-Cluster verwenden können. Durch die Verteilung Ihrer Cluster-Instances auf Availability Zones stellen Sie sicher, dass Instances in Ihrem Cluster für den unwahrscheinlichen Fall eines Ausfalls der Availability Zone verfügbar sind.

Voraussetzungen für das Subnetz

Wenn Sie einen Amazon DocumentDB-Cluster erstellen, müssen Sie eine VPC und die entsprechende Subnetzgruppe innerhalb dieser VPC auswählen, um Ihren Cluster zu starten. Subnetze bestimmen die Availability Zone und den IP-Bereich innerhalb dieser Availability Zone, die Sie zum Starten einer Instance verwenden möchten. Im Rahmen dieser Ausführungen werden die Begriffe Subnetz und Availability Zone synonym verwendet. Eine Subnetzgruppe ist ein benannter Satz von Subnetzen (oder Availability Zones). Mit einer Subnetzgruppe können Sie die Availability Zones angeben, die Sie zum Starten von Amazon DocumentDB DocumentDB-Instances verwenden möchten. So wird beispielsweise in einem Cluster mit drei Instances für eine hohe Verfügbarkeit empfohlen, dass jede dieser Instances in separaten Availability Zones bereitgestellt wird. Wenn eine einzelne Availability Zones ausfällt, wirkt sich dies nur auf eine einzelne Instance aus.

Amazon DocumentDB DocumentDB-Instances können derzeit in bis zu drei Availability Zones bereitgestellt werden. Selbst wenn eine Subnetzgruppe mehr als drei Subnetze hat, können Sie nur drei dieser Subnetze verwenden, um einen Amazon DocumentDB-Cluster zu erstellen. Daher wird empfohlen, dass Sie beim Anlegen einer Subnetzgruppe nur die drei Subnetze auswählen, in denen Sie Ihre Instances bereitstellen möchten. In USA Ost (Nord-Virginia) kann Ihre Subnetzgruppe sechs Subnetze (oder Availability Zones) haben. Wenn jedoch ein Amazon DocumentDB-Cluster bereitgestellt wird, wählt Amazon DocumentDB drei der Availability Zones aus, die es für die Bereitstellung von Instances verwendet.

Nehmen wir beispielsweise an, dass Amazon DocumentDB beim Erstellen eines Clusters die Availability Zones {1A, 1B und 1C} auswählt. Wenn Sie nun versuchen, eine Instance in Availability Zone {1D} zu erstellen, schlägt der API-Aufruf fehl. Wenn Sie sich jedoch dafür entscheiden, eine Instance zu erstellen, ohne eine bestimmte Availability Zone anzugeben, wählt Amazon DocumentDB in Ihrem Namen eine Availability Zone aus. Amazon DocumentDB verwendet einen Algorithmus für den Lastenausgleich der Instances in den Availability Zones, um Ihnen zu helfen, eine hohe Verfügbarkeit zu erreichen. Wenn beispielsweise drei Instances bereitgestellt werden, werden diese standardmäßig über drei Availability Zones verteilt und nicht alle in einer einzigen Availability Zone bereitgestellt.

Empfehlungen:

- Wenn Sie keinen speziellen Grund haben, legen Sie immer eine Subnetzgruppe mit drei Subnetzen an. Auf diese Weise wird sichergestellt, dass Cluster mit drei oder mehr Instances eine höhere Verfügbarkeit erreichen können, da Instances über drei Availability Zones bereitgestellt werden.
- Verteilen Sie Instances immer über mehrere Availability Zones, um eine hohe Verfügbarkeit zu erreichen. Platzieren Sie niemals alle Instances für einen Cluster in einer einzigen Availability Zone.
- Da Failover-Ereignisse jederzeit auftreten können, sollten Sie nicht davon ausgehen, dass sich eine primäre Instance oder Replikat-Instance immer in einer bestimmten Availability Zone befinden.

Zusätzliche Voraussetzungen

Im Folgenden sind einige zusätzliche Voraussetzungen für die Erstellung eines Amazon DocumentDB-Clusters aufgeführt:

- Wenn Sie eine Verbindung AWS mit AWS Identity and Access Management (IAM-) Anmeldeinformationen herstellen, muss Ihr IAM-Konto über IAM-Richtlinien verfügen, die die Berechtigungen gewähren, die für die Ausführung von Amazon DocumentDB DocumentDB-Vorgängen erforderlich sind.

Wenn Sie ein IAM-Konto für den Zugriff auf die Amazon DocumentDB DocumentDB-Konsole verwenden, müssen Sie sich zunächst AWS Management Console mit Ihrem IAM-Konto bei der anmelden. Rufen Sie dann die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb auf.

- Wenn Sie die Konfigurationsparameter für Ihren Cluster anpassen möchten, müssen Sie eine Cluster-Parametergruppe und eine Parametergruppe mit den erforderlichen Parametereinstellungen festlegen. Informationen zum Erstellen oder Ändern einer Cluster-Parametergruppe oder Parametergruppe finden Sie unter Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen.
- Sie müssen die TCP/IP Portnummer ermitteln, die Sie für Ihren Cluster angeben möchten. Die Firewalls einiger Unternehmen blockieren Verbindungen zu den Standardports für Amazon DocumentDB. Wenn die Firewall Ihres Unternehmens den Standard-Port blockiert, wählen Sie einen anderen Port für Ihr Cluster aus. Alle Instances in einem Cluster verwenden denselben Port.

Erstellen eines Clusters und einer primären Instance mit dem AWS Management Console

Die folgenden Verfahren beschreiben, wie Sie die Konsole verwenden, um einen Amazon DocumentDB-Cluster mit einer oder mehreren Instances zu starten.

Erstellen Sie einen Cluster: Verwenden Sie die Standardeinstellungen

Um einen Cluster mit Instanzen zu erstellen, verwenden Sie die Standardeinstellungen AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wenn Sie Ihren Cluster in einer AWS-Region anderen Region als der Region USA Ost (Nord-Virginia) erstellen möchten, wählen Sie die Region aus der Liste oben rechts in der Konsole aus.
3. Wählen Sie im Navigationsbereich Clusters (Cluster) und dann Create (Erstellen).

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

4. Füllen Sie auf der Seite Amazon DocumentDB-Cluster erstellen den Bereich Konfiguration aus.

   a. Cluster-ID — Akzeptieren Sie den von Amazon DocumentDB angegebenen Namen, oder geben Sie einen Namen für Ihren Cluster ein, zum Beispiel. **`sample-cluster`**

      Einschränkungen bei der Benennung von Clustern:

      - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Cluster in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.

   b. Engine-Version — Akzeptieren Sie die Standard-Engine-Version 5.0.0 oder wählen Sie optional 4.0.0 oder 3.6.0.

   c. Instanzklasse — Akzeptieren Sie die Standardklasse`db.r5.large`, oder wählen Sie die gewünschte Instanzklasse aus der Liste aus.

   d. Anzahl der Instanzen — Wählen Sie in der Liste die Anzahl der Instances aus, die Sie mit diesem Cluster erstellen möchten. Die erste Instance ist die primäre Instance. Alle anderen Instances sind schreibgeschützte Replikat-Instances. Sie können später Instances hinzufügen und löschen. Standardmäßig wird ein Amazon DocumentDB-Cluster mit drei Instances (einer primären und zwei Replikaten) gestartet.

5. Füllen Sie den Abschnitt Cluster-Speicherkonfiguration aus.

   Wählen Sie entweder Amazon DocumentDB Standard (Standard) oder Amazon DocumentDB I/O-Optimized. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Speicherkonfigurationen.

6. Vervollständigen Sie den Bereich Authentication (Authentifizierung).

   a. Benutzername — Geben Sie einen Namen für den Hauptbenutzer ein. Um sich bei Ihrem Cluster anzumelden, müssen Sie den primären Benutzernamen verwenden.

      Einschränkungen bei der Benennung primärer Benutzer:

      - Die Länge beträgt [1—63] alphanumerische Zeichen.
      - Muss mit einem Buchstaben beginnen.
      - Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

b. Wählen Sie eine der folgenden Passwortoptionen:

- Verwaltet in AWS Secrets Manager — Wählen Sie diese Option, wenn Sie Ihr primäres Benutzerkennwort automatisch verwalten möchten AWS Secrets Manager .

  Wenn Sie diese Option wählen, konfigurieren Sie den KMS-Schlüssel, indem Sie entweder Ihren eigenen Schlüssel erstellen oder einen Schlüssel verwenden, den Secrets Manager erstellt.

- Selbstverwaltet — Wählen Sie diese Option, wenn Sie Ihr primäres Benutzerkennwort selbst verwalten möchten. Wenn Sie diese Option wählen, geben Sie ein Passwort für den Hauptbenutzer ein und bestätigen Sie es dann. Um sich bei Ihrem Cluster anzumelden, müssen Sie das Passwort für den Hauptbenutzer verwenden.

  Passwortbeschränkungen:

  - Länge beträgt [8–100] druckbare ASCII-Zeichen.
  - Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
    - **/** (Schrägstrich)
    - **"** (doppeltes Anführungszeichen)
    - **@** ('At'-Symbol)

7. Wählen Sie am unteren Rand der Seite eine der folgenden Optionen:

- Um den Cluster jetzt zu erstellen, wählen Sie Create Cluster (Cluster erstellen) aus.
- Wenn der Cluster nicht neu erstellt werden soll, wählen Sie Cancel (Abbrechen) aus.
- Um den Cluster vor der Erstellung weiter zu konfigurieren, wählen Sie Show additional configurations (Weitere Konfigurationen anzeigen) aus und fahren dann mit Einen Cluster erstellen: zusätzliche Konfigurationen fort.

  Im Bereich Additional Configurations (Zusätzliche Konfigurationen) sind die folgenden Konfigurationen zu finden:

  - Netzwerkeinstellungen — Standardmäßig wird die `default` VPC-Sicherheitsgruppe verwendet.
  - Clusteroptionen — Standardmäßig wird der Port 27017 und die Standardparametergruppe verwendet.

- Verschlüsselung — Standardmäßig wird die Verschlüsselung mithilfe des `(default) aws/rds` Schlüssels aktiviert.

  > **Important**
  >
  > Die Verschlüsselung eines einmal verschlüsselten Clusters kann nicht wieder aufgehoben werden.

- Backup — Standardmäßig werden Backups für einen Tag aufbewahrt und Amazon DocumentDB das Backup-Fenster wählen lassen.
- Protokollexporte — Standardmäßig werden Audit-Logs nicht in Logs exportiert CloudWatch .
- Wartung — Standardmäßig lässt Amazon DocumentDB das Wartungsfenster wählen.
- Löschschutz — Schützt Ihren Cluster vor versehentlichem Löschen. Cluster, die mit der Konsole erstellt wurden, sind standardmäßig aktiviert.

Wenn Sie die Standardeinstellungen jetzt übernehmen, können Sie die meisten davon später durch Bearbeitung des Clusters wieder ändern.

8. Aktivieren Sie eingehende Verbindungen für die Sicherheitsgruppe Ihres Clusters.

   Wenn Sie die Standardeinstellungen für Ihr Cluster nicht geändert haben, wurde mithilfe der Standard-Sicherheitsgruppe ein Cluster für die Standard-VPC in der angegebenen Region erstellt. Um eine Verbindung zu Amazon DocumentDB herzustellen, müssen Sie eingehende Verbindungen auf Port 27017 (oder dem Port Ihrer Wahl) für die Sicherheitsgruppe Ihres Clusters aktivieren.

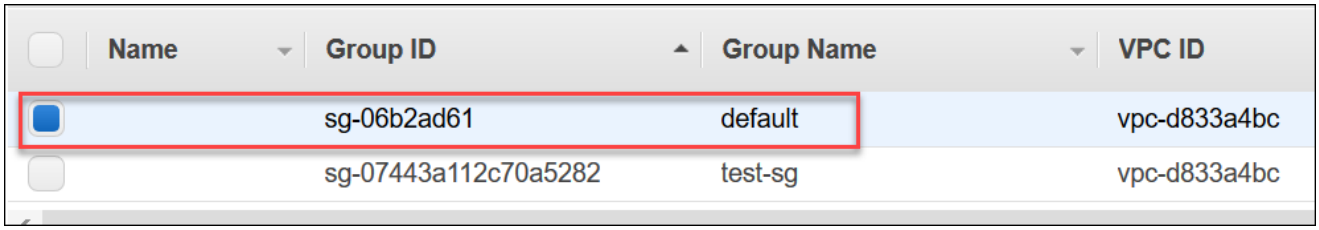
   So fügen Sie eine eingehende Verbindung zur SicherheitsgruppeIhres Clusters hinzu:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.

   b. Wählen Sie im Abschnitt Resources (Ressourcen) des Hauptfensters die Option Security groups (Sicherheitsgruppen).



c. Wählen Sie in der Liste der Sicherheitsgruppen die Sicherheitsgruppe aus, die Sie bei der Erstellung Ihres Cluster verwendet haben (höchstwahrscheinlich handelt es sich dabei um die Standard-Sicherheitsgruppe), und wählen Sie das Feld links neben dem Name der Sicherheitsgruppe aus.



d. Wählen Sie im Menü Aktionen die Option Eingangsregeln bearbeiten aus. Wählen Sie dann die Regeleinschränkungen aus oder geben Sie diese ein.

   i. Typ — Wählen Sie aus der Liste das Protokoll aus, das für den Netzwerkverkehr geöffnet werden soll.

   ii. Protokoll — Wählen Sie aus der Liste den Protokolltyp aus.

   iii. Portbereich — Geben Sie für eine benutzerdefinierte Regel eine Portnummer oder einen Portbereich ein. Stellen Sie sicher, dass die Portnummer bzw. der Portbereich den Port enthält, den Sie beim Erstellen des Clusters angegeben haben (Standard: 27017).

   iv. Quelle — Gibt den Datenverkehr an, der Ihre Instance erreichen kann. Wählen Sie die Datenverkehrquelle aus der Liste aus. Wenn Sie Custom (Benutzerdefiniert) auswählen, geben Sie eine einzelne IP-Adresse oder einen IP-Adressbereich CIDR-Notation an (z. B. 203.0.113.5/32).

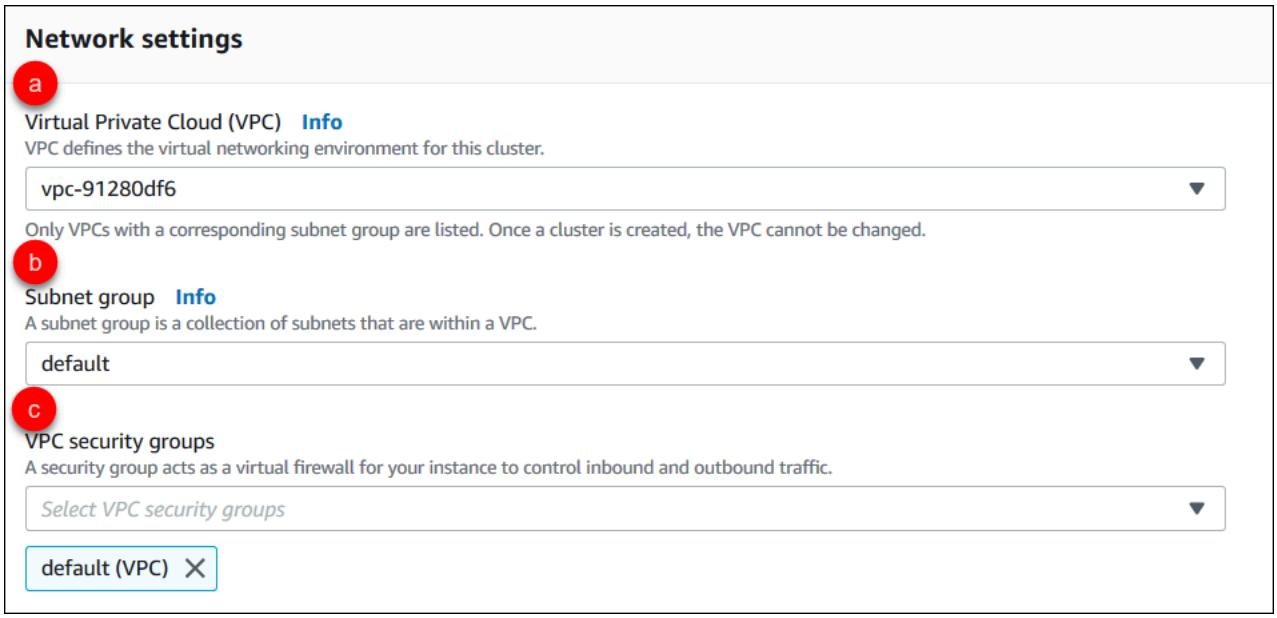
   v. Beschreibung — Geben Sie eine Beschreibung für diese Regel ein.

   vi. Wenn Sie die Regel fertig erstellt haben, wählen Sie Save (Speichern).

Einen Cluster erstellen: zusätzliche Konfigurationen

Wenn Sie die Standardeinstellungen für Ihren Cluster übernehmen möchten, können Sie die folgenden Schritte überspringen und Create cluster (Cluster erstellen) auswählen.

1. Vervollständigen Sie den Bereich Network settings (Netzwerkeinstellungen).



   a. Virtual Private Cloud (VPC) — Wählen Sie in der Liste die Amazon-VPC aus, in der Sie diesen Cluster starten möchten.

   b. Subnetzgruppe — Wählen Sie in der Liste die Subnetzgruppe aus, die Sie für diesen Cluster verwenden möchten.

   c. VPC-Sicherheitsgruppen — Wählen Sie in der Liste die VPC-Sicherheitsgruppe für diesen Cluster aus.

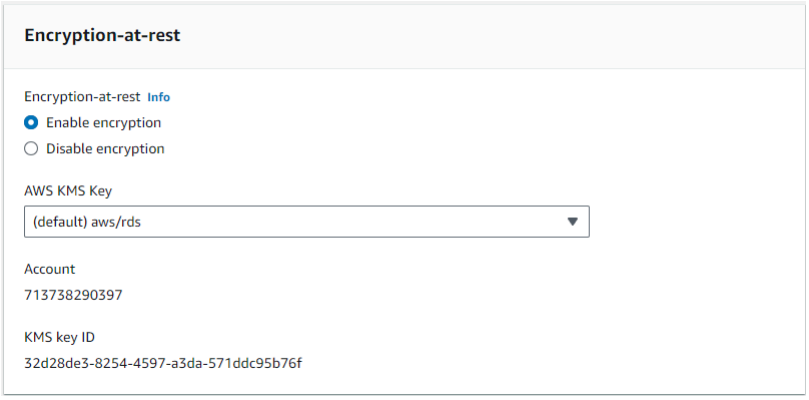
2. Vervollständigen Sie den Bereich Cluster options (Cluster-Optionen).



a. Datenbankport — Verwenden Sie die Aufwärts- und Abwärtspfeile, um den TCP/IP Port festzulegen, über den Anwendungen eine Verbindung zu Ihrer Instance herstellen.

b. Cluster-Parametergruppe — Wählen Sie in der Liste der Parametergruppen die Cluster-Parametergruppe für diesen Cluster aus.

3. Vervollständigen Sie den Bereich Encryption (Verschlüsselung).



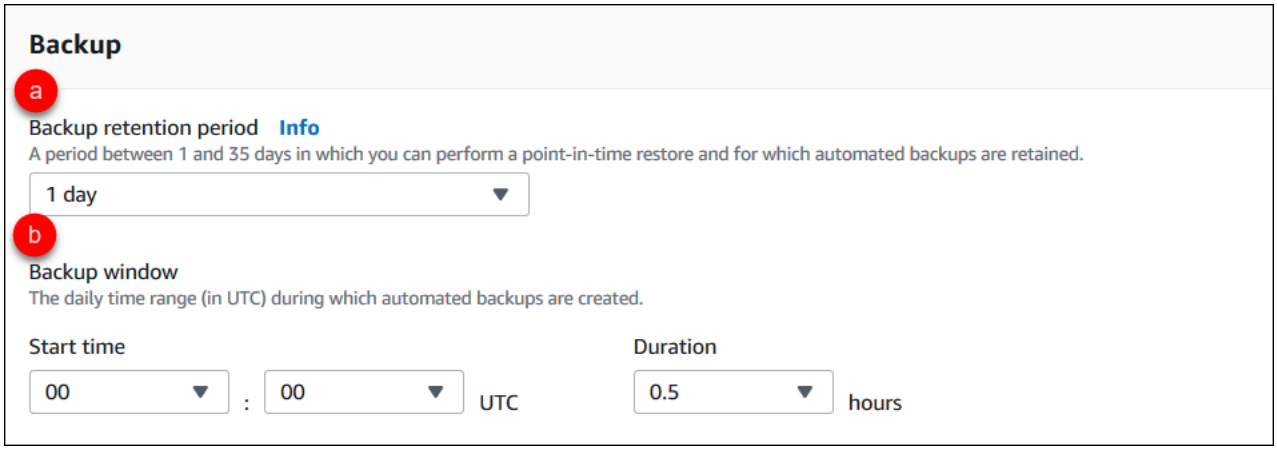
a. E ncryption-at-rest — Wählen Sie eine der folgenden Optionen:

- Verschlüsselung aktivieren — Standard. Alle Daten im Ruhezustand werden verschlüsselt. Wenn Sie sich für die Verschlüsselung Ihrer Daten entscheiden, können Sie diese Aktion nicht rückgängig machen.
- Verschlüsselung deaktivieren — Ihre Daten sind nicht verschlüsselt.

b. AWS KMS-Schlüssel — Dieser Schlüssel ist nur verfügbar, wenn Sie Ihre Daten verschlüsseln. Wählen Sie aus der Liste den Schlüssel aus, den Sie für die Verschlüsselung der Daten in diesem Cluster verwenden möchten. Der Standardwert ist `(default) aws/rds`.

Wenn Sie Enter a key ARN (Einen Schlüssel-ARN eingeben) auswählen, müssen Sie einen Amazon-Ressourcenname (ARN) für den Schlüssel eingeben.

4. Vervollständigen Sie den Bereich Backup.



a. Aufbewahrungszeitraum für Backup — Wählen Sie in der Liste die Anzahl der Tage aus, für die automatische Backups dieses Clusters aufbewahrt werden sollen, bevor sie gelöscht werden.

b. Backup-Fenster — Legen Sie die tägliche Uhrzeit und Dauer fest, während der Amazon DocumentDB Backups dieses Clusters erstellen soll.

   i. Startzeit — Wählen Sie in der ersten Liste die Startzeit (UTC) für den Start Ihrer automatischen Backups aus. Wählen Sie in der zweiten Liste die Minute für den Beginn der automatischen Backups aus.

   ii. Dauer — Wählen Sie in der Liste die Anzahl der Stunden aus, die für die Erstellung automatischer Backups reserviert werden sollen.

5. Füllen Sie den Bereich Protokollexporte aus, indem Sie die Protokolltypen auswählen, die Sie in CloudWatch Protokolle exportieren möchten.



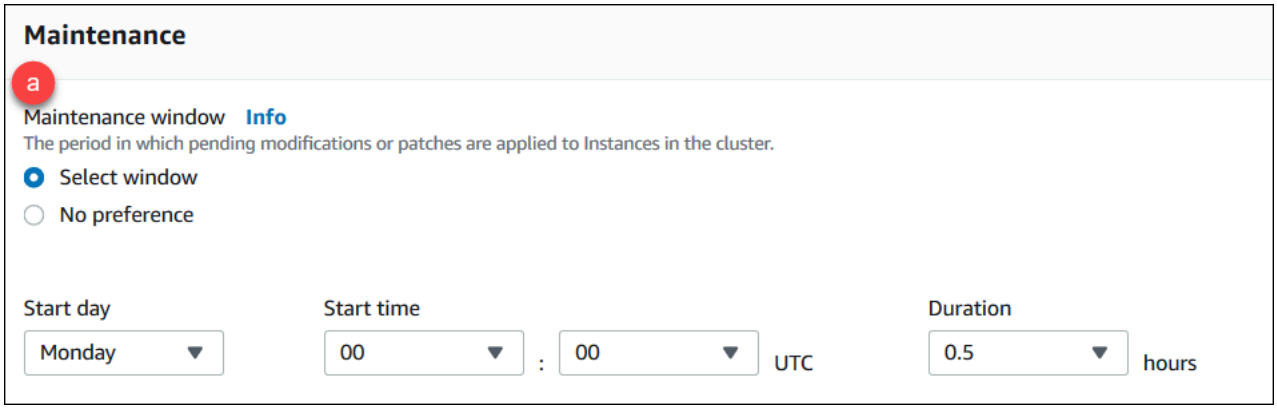
- Audit-Logs — Wählen Sie diese Option, um den Export von Audit-Logs nach Amazon CloudWatch Logs zu ermöglichen. Wenn Sie Audit logs (Auditprotokolle), auswählen, müssen Sie `audit_logs` in der benutzerdefinierten Parametergruppe des Clusters aktivieren. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.
- Profiler-Protokolle — Wählen Sie diese Option, um den Export von Operation-Profiler-Protokollen nach Amazon Logs zu aktivieren. CloudWatch Wenn Sie Profiler logs (Profiler-

Protokolle) auswählen, müssen Sie auch die folgenden Parameter in der benutzerdefinierten Parametergruppe des Clusters ändern:

- `profiler`— Eingestellt auf. `enabled`
- `profiler_threshold_ms`— Auf einen Wert setzen, `[0-INT_MAX]` um den Schwellenwert für Profilerstellungsvorgänge festzulegen.
- `profiler_sampling_rate`— Legen Sie einen Wert fest, um den Prozentsatz langsamer Vorgänge festzulegen`[0.0-1.0]`, für den ein Profil erstellt werden soll.

Weitere Informationen finden Sie unter Profilierung von Amazon DocumentDB DocumentDB-Vorgängen.

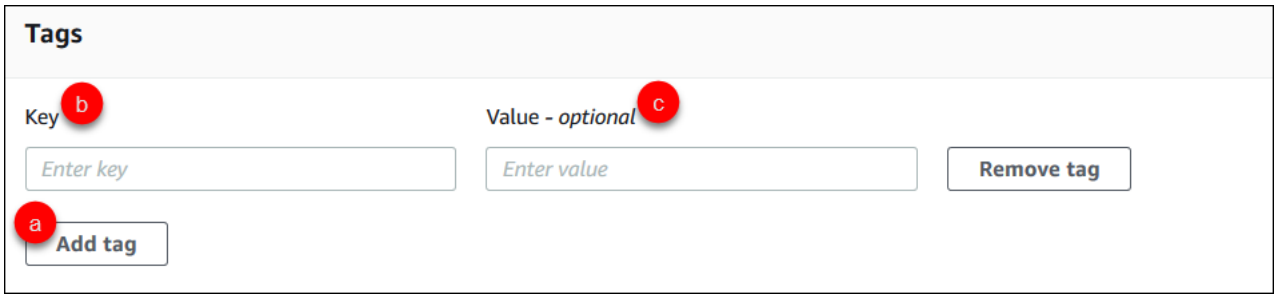
6. Schließen Sie den Bereich Maintenance (Wartung) ab.



- Wählen Sie eine der folgenden Optionen aus:
  - Fenster auswählen — Sie können den Wochentag, die UTC-Startzeit und die Dauer angeben, für die Amazon DocumentDB Wartungsarbeiten an Ihrem Cluster durchführt.
    a. Starttag — Wählen Sie in der Liste den Wochentag aus, an dem die Cluster-Wartung gestartet werden soll.
    b. Startzeit — Wählen Sie in den Listen die Stunde und Minute (UTC) aus, um mit der Wartung zu beginnen.
    c. Dauer — Wählen Sie in der Liste aus, wie viel Zeit für die Cluster-Wartung reserviert werden soll. Wenn die Wartung nicht innerhalb der angegebenen Zeit abgeschlossen werden kann, wird der Wartungsprozess über die angegebene Zeit hinaus bis zum Abschluss fortgesetzt.

- Keine Präferenz — Amazon DocumentDB wählt den Wochentag, die Startzeit und die Dauer für die Durchführung von Wartungsarbeiten.

7. Wenn Sie diesem Cluster ein oder mehrere Tags hinzufügen möchten, stellen Sie den Bereich Tags fertig.



Wiederholen Sie die folgenden Schritte für alle Tags, die Sie dem Cluster hinzufügen möchten. Sie können bis zu 10 in einem Cluster haben.

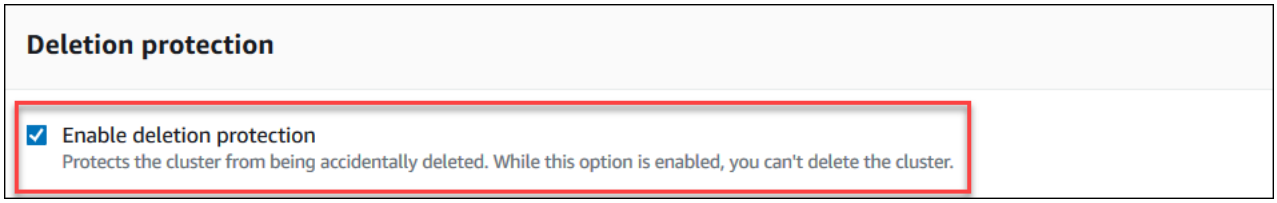
a. Wählen Sie Tags hinzufügen aus.

b. Geben Sie den Schlüssel des Tags ein.

c. Optional können Sie auch den Wert des Tags eingeben.

Klicken Sie zum Entfernen eines Tags auf Tag entfernen.

8. Der Löschschutz ist standardmäßig aktiviert, wenn Sie ein Cluster mit der Konsole erstellen. Deaktivieren Sie zum Ausschalten des Löschschutzes die Option Löschschutz aktivieren. Wenn diese Option aktiviert ist, verhindert der Löschschutz das Löschen eines Clusters. Wenn Sie einen löschgeschützten Cluster löschen möchten, müssen Sie zuerst den Cluster-Löschschutz deaktivieren.



Weitere Informationen über den Löschschutz finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

9. Wählen Sie zum Erstellen des Clusters Cluster erstellen aus. Wählen Sie andernfalls Abbrechen.

Erstellen eines Clusters mit dem AWS CLI

Die folgenden Verfahren beschreiben, wie Sie den verwenden, um einen Amazon DocumentDB-Cluster AWS CLI zu starten und ein Amazon DocumentDB DocumentDB-Replikat zu erstellen.

Parameter

- **`--db-cluster-identifier`**—Erforderlich. Eine Zeichenfolge in Kleinbuchstaben, die diesen Cluster identifiziert.

  Einschränkungen bei der Benennung von Clustern:

  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster (in Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

- **`--engine`**—Erforderlich. Der Wert muss **`docdb`** sein.
- **`--deletion-protection | --no-deletion-protection`**— Fakultativ. Der aktivierte Löschschutz verhindert, dass ein Cluster gelöscht wird. Wenn Sie die verwenden AWS CLI, ist der Löschschutz standardmäßig deaktiviert.

  Weitere Informationen über den Löschschutz finden Sie unter Löschen eines Amazon DocumentDB-Clusters.
- **`--storage-type standard | iopt1`**— Fakultativ. Standard: **`standard`**. Die Speicherkonfiguration des Clusters. Gültige Werte sind `standard` (Standard) oder `iopt1` (I/O-optimiert).
- **`--master-username`**—Erforderlich. Der Benutzername für die Authentifizierung des Benutzers.

  Namenseinschränkungen für Benutzer:

  - Die Länge beträgt [1 bis 63] alphanumerische Zeichen.
  - Muss mit einem Buchstaben beginnen.
  - Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

- **`--master-user-password`**— Optional. Das Passwort für die Authentifizierung des Benutzers.

Einschränkungen für Hauptpasswort:

- Länge beträgt [8–100] druckbare ASCII-Zeichen.
- Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
  - **/** (Schrägstrich)
  - **"** (doppeltes Anführungszeichen)
  - **@** ('At'-Symbol)

- **`--manage-master-user-password`**—Fakultativ. Amazon DocumentDB generiert das Masterbenutzerkennwort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.

Weitere Parameter finden Sie unter CreateDBCluster.

Um einen Amazon DocumentDB-Cluster mit dem AWS CLI

Um einen Amazon DocumentDB-Cluster zu erstellen, rufen Sie den auf `create-db-cluster` AWS CLI. Der folgende AWS CLI Befehl erstellt einen Amazon DocumentDB-Cluster `sample-cluster` mit dem Namen und aktiviertem Löschschutz. Weitere Informationen zum Löschschutz finden Sie unterLöschen eines Amazon DocumentDB-Clusters.

Außerdem `--engine-version` ist dies ein optionaler Parameter, der standardmäßig die neueste Hauptversion der Engine verwendet. Die aktuelle Version der Hauptengine ist 5.0.0. Wenn neue Hauptversionen der Engine veröffentlicht werden, `--engine-version` wird die Standard-Engine-Version für aktualisiert, sodass sie die letzte Hauptversion der Engine wiedergibt. Daher empfehlen wir für Produktionsworkloads, insbesondere solche, die von Skripten, Automatisierung oder AWS CloudFormation Vorlagen abhängig sind, die Angabe der `--engine-version` beabsichtigten Hauptversion ausdrücklich.

> **Note**
>
> Wenn ein `db-subnet-group-name` oder nicht angegeben `vpc-security-group-id` ist, verwendet Amazon DocumentDB die Standard-Subnetzgruppe und die Amazon VPC-Sicherheitsgruppe für die angegebene Region.

Für Linux, macOS oder Unix:

```
aws docdb create-db-cluster \
      --db-cluster-identifier sample-cluster \
      --engine docdb \
      --engine-version 5.0.0 \
      --deletion-protection \
      --master-username masteruser \
      --master-user-password password
```

Für Windows:

```
aws docdb create-db-cluster ^
      --db-cluster-identifier sample-cluster ^
      --engine docdb ^
      --engine-version 5.0.0 ^
      --deletion-protection ^
      --master-username masteruser ^
      --master-user-password password
```

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

```
{
    "DBCluster": {
        "StorageEncrypted": false,
        "DBClusterMembers": [],
        "Engine": "docdb",
        "DeletionProtection" : "enabled",
        "ClusterCreateTime": "2018-11-26T17:15:19.885Z",
        "DBSubnetGroup": "default",
        "EngineVersion": "5.0.0",
        "MasterUsername": "masteruser",
        "BackupRetentionPeriod": 1,
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "DBClusterIdentifier": "sample-cluster",
        "MultiAZ": false,
        "DBClusterParameterGroup": "default.docdb5.0",
        "PreferredBackupWindow": "09:12-09:42",
        "DbClusterResourceId": "cluster-KQSGI4MHU4NTDDRVNLNTU7XVAY",
        "PreferredMaintenanceWindow": "tue:04:17-tue:04:47",
        "Port": 27017,
        "Status": "creating",
        "ReaderEndpoint": "sample-cluster.cluster-ro-sfcrlcjcoroz.us-
east-1.docdb.amazonaws.com",
```

```
        "AssociatedRoles": [],
        "HostedZoneId": "ZNKXTT8WH85VW",
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1e"
        ],
        "Endpoint": "sample-cluster.cluster-sfcrlcjcoroz.us-east-1.docdb.amazonaws.com"
    }
}
```

Die Erstellung des Clusters dauert mehrere Minuten. Sie können das AWS Management Console oder verwenden AWS CLI , um den Status Ihres Clusters zu überwachen. Weitere Informationen finden Sie unter Den Status eines Amazon DocumentDB-Clusters überwachen.

> **Important**
>
> Wenn Sie den verwenden AWS CLI , um einen Amazon DocumentDB-Cluster zu erstellen, werden keine Instances erstellt. Daher müssen Sie explizit eine primäre Instance und alle benötigten Replikat-Instances anlegen. Sie können entweder die Konsole oder AWS CLI zum Erstellen der Instances verwenden. Weitere Informationen finden Sie unter Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster.

Weitere Informationen finden Sie `CreateDBCluster`in der Amazon DocumentDB DocumentDB-API-Referenz.

Beschreibung von Amazon DocumentDB-Clustern

Sie können entweder die Amazon DocumentDB Management Console oder die verwenden, AWS CLI um Details wie Verbindungsendpunkte, Sicherheitsgruppen und Parametergruppen zu Ihren Amazon DocumentDB-Clustern anzuzeigen. VPCs

Weitere Informationen finden Sie hier:

- Den Status eines Amazon DocumentDB-Clusters überwachen
- Die Endpunkte eines Clusters finden

Using the AWS Management Console

Gehen Sie wie folgt vor, um die Details eines bestimmten Amazon DocumentDB-Clusters mithilfe der Konsole anzuzeigen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, dessen Details Sie sehen möchten. Die Informationen über den Cluster sind in die folgenden Gruppierungen unterteilt:

   - Zusammenfassung — Allgemeine Informationen über den Cluster, einschließlich der Engine-Version, des Cluster-Status, ausstehender Wartungsarbeiten und des Status der zugehörigen Parametergruppe.
   - Konnektivität und Sicherheit — Der Abschnitt Connect listet Verbindungsendpunkte auf, um mit der Mongo-Shell oder mit einer Anwendung eine Verbindung zu diesem Cluster herzustellen. Im Abschnitt Sicherheitsgruppen werden die Sicherheitsgruppen aufgeführt, die diesem Cluster zugeordnet sind, sowie ihre VPC-ID und Beschreibungen.
   - Konfiguration — Im Abschnitt Cluster-Details werden Details zum Cluster aufgeführt, einschließlich des Amazon-Ressourcennamens (ARN), des Endpunkts und der Parametergruppe des Clusters. Außerdem werden die Sicherungsinformationen, Wartungsdetails sowie Sicherheits- und Netzwerkeinstellungen des Clusters aufgelistet. Im Abschnitt Cluster-Instances werden die Instances aufgeführt, die zu diesem Cluster gehören, mit dem Status der einzelnen Instance-Rollen und Cluster-Parametergruppen.
   - Überwachung — Die Amazon CloudWatch Logs-Metriken für diesen Cluster. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.

- Ereignisse und Tags — Im Abschnitt Aktuelle Ereignisse werden die jüngsten Ereignisse für diesen Cluster aufgeführt. Amazon DocumentDB zeichnet Ereignisse auf, die sich auf Ihre Cluster, Instances, Snapshots, Sicherheitsgruppen und Cluster-Parametergruppen beziehen. Zu diesen Informationen gehören Datum, Uhrzeit und Nachricht, die jedem Ereignis zugeordnet sind. Der Abschnitt Tags listet die an diesen Cluster angehängten Tags auf.

Using the AWS CLI

Um die Details Ihrer Amazon DocumentDB-Cluster mithilfe von anzuzeigen AWS CLI, verwenden Sie den `describe-db-clusters` Befehl, wie in den folgenden Beispielen gezeigt. Weitere Informationen finden Sie `DescribeDBClusters`in der Amazon DocumentDB Resource Management API-Referenz.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen wie Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird. Der `filterName=engine,Values=docdb` Filterparameter gibt nur Amazon DocumentDB-Cluster zurück.

Example

Beispiel 1: Alle Amazon DocumentDB-Cluster auflisten

Der folgende AWS CLI Code listet die Details für alle Amazon DocumentDB-Cluster in einer Region auf.

```
aws docdb describe-db-clusters --filter Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusters": [
        {
            "AvailabilityZones": [
                "us-east-1c",
```

```
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-1",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-2",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        },
        {
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1b",
                "us-east-1a"
            ],
            "BackupRetentionPeriod": 1,
            "DBClusterIdentifier": "sample-cluster-3",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            ...
        }
    ]
}
```

Example

Beispiel 2: Alle Details für einen bestimmten Amazon DocumentDB-Cluster auflisten

Der folgende AWS CLI Code listet die Details für den Cluster `sample-cluster` auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
   --filter Name=engine,Values=docdb \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb describe-db-clusters ^
   --filter Name=engine,Values=docdb ^
   --db-cluster-identifier sample-cluster
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBClusters": [
        {
            "AllocatedStorage": 1,
            "AvailabilityZones": [
                "us-east-1c",
                "us-east-1a",
                "us-east-1d"
            ],
            "BackupRetentionPeriod": 2,
            "DBClusterIdentifier": "sample-cluster",
            "DBClusterParameterGroup": "sample-parameter-group",
            "DBSubnetGroup": "default",
            "Status": "available",
            "EarliestRestorableTime": "2023-11-07T22:34:08.148000+00:00",
            "Endpoint": "sample-cluster.node.us-east-1.amazon.com",
            "ReaderEndpoint": "sample-cluster.node.us-east-1.amazon.com",
            "MultiAZ": false,
            "Engine": "docdb",
            "EngineVersion": "5.0.0",
            "LatestRestorableTime": "2023-11-10T07:21:16.772000+00:00",
            "Port": 27017,
            "MasterUsername": "chimeraAdmin",
            "PreferredBackupWindow": "22:22-22:52",
            "PreferredMaintenanceWindow": "sun:03:01-sun:03:31",
            "ReadReplicaIdentifiers": [],
            "DBClusterMembers": [
                {
                    "DBInstanceIdentifier": "sample-instance-1",
```

```
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },
                {
                    "DBInstanceIdentifier": "sample-instance-2",
                    "IsClusterWriter": true,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "PromotionTier": 1
                },

            ],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-9084c2ec",
                    "Status": "active"
                }
            ],
            "HostedZoneId": "Z06853723JYKYBXTJ49RB",
            "StorageEncrypted": false,
            "DbClusterResourceId": "cluster-T4LGLANHVAPGQYYULWUDKLVQL4",
            "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster",
            "AssociatedRoles": [],
            "IAMDatabaseAuthenticationEnabled": false,
            "ClusterCreateTime": "2023-11-06T18:05:41.568000+00:00",
            "EngineMode": "provisioned",
            "DeletionProtection": false,
            "HttpEndpointEnabled": false,
            "CopyTagsToSnapshot": false,
            "CrossAccountClone": false,
            "DomainMemberships": [],
            "TagList": [],
            "StorageType": "iopt1",
            "AutoMinorVersionUpgrade": false,
            "NetworkType": "IPV4",
            "IOOptimizedNextAllowedModificationTime":
 "2023-12-07T18:05:41.580000+00:00"
        }
    ]
}
```

Example

Beispiel 3: Spezifische Details für einen Amazon DocumentDB-Cluster auflisten

Um eine Teilmenge der Cluster-Details mithilfe von aufzulisten AWS CLI, fügen Sie eine hinzu, die `--query` angibt, welche Cluster-Mitglieder der `describe-db-clusters` Vorgang auflisten soll. Der `--db-cluster-identifier`-Parameter ist der Bezeichner für den jeweiligen Cluster, dessen Details Sie anzeigen möchten. Weitere Informationen zu Abfragen finden Sie unter So filtern Sie die Ausgabe mit der *`--query`* Option im AWS Command Line Interface Benutzerhandbuch.

Das folgende Beispiel listet die Instances in einem Amazon DocumentDB-Cluster auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --filter Name=engine,Values=docdb \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterMembers]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --filter Name=engine,Values=docdb ^
    --db-cluster-identifier sample-cluster ^
    --query 'DBClusters[*].[DBClusterMembers]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": false,
                "DBClusterParameterGroupStatus": "in-sync",
```

```
                "PromotionTier": 1
            }
        ]
    ]
]
```

Ändern eines Amazon DocumentDB-Clusters

Nur Cluster, die sich im Status available befinden, können geändert werden. Ein angehaltener Cluster kann nicht geändert werden. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis der Cluster verfügbar wird, und nehmen Sie dann die gewünschten Änderungen vor. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster stoppen und starten.

Using the AWS Management Console

Gehen Sie wie folgt vor, um einen bestimmten Amazon DocumentDB-Cluster mithilfe der Konsole zu ändern.

So ändern Sie einen Amazon DocumentDB-Cluster

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Geben Sie den Cluster an, den Sie ändern möchten, indem Sie die Schaltfläche links neben dem Namen des Clusters auswählen.
4. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
5. Nehmen Sie im Bereich Modify Cluster: <cluster-name> (Cluster ändern: <cluster-name>) die gewünschten Änderungen vor. Sie können in den folgenden Bereichen Änderungen durchführen:

- Clusterspezifikationen — Der Name, die Sicherheitsgruppen und die Verwaltung der Anmeldeinformationen des Clusters.
- Cluster-Speicherkonfiguration — Der Datenspeichermodus des Clusters. Wählen Sie zwischen Standard- und I/O-optimierter Konfiguration.
- Clusteroptionen — Der Port und die Parametergruppe des Clusters.
- Backup — Der Aufbewahrungszeitraum für Backups und das Backup-Fenster des Clusters.
- Protokollexporte — Aktiviert oder deaktiviert den Export von Audit- oder Profiler-Protokollen.
- Wartung — Legt das Wartungsfenster des Clusters fest.
- Löschschutz — Aktiviert oder deaktiviert den Löschschutz auf dem Cluster. Der Löschschutz ist standardmäßig aktiviert.

6. Wenn Sie fertig sind, wählen Sie Continue (Weiter) aus, um eine Zusammenfassung Ihrer Änderungen anzuzeigen.
7. Wenn Sie mit den Änderungen zufrieden sind, können Sie Modify cluster (Cluster ändern) auswählen, um den Cluster zu ändern. Alternativ können Sie Back (Zurück) oder Cancel (Abbrechen) wählen, um die Änderungen zu bearbeiten oder abzubrechen.

Es dauert einige Minuten, bis Ihre Änderungen übernommen werden. Sie können den Cluster nur verwenden, wenn sein Status available ist. Sie können den Status des Clusters über die Konsole oder AWS CLIüberwachen. Weitere Informationen finden Sie unter Den Status eines Amazon DocumentDB-Clusters überwachen.

Using the AWS CLI

Verwenden Sie die `modify-db-cluster`-Operation, um den angegebenen Cluster mit der AWS CLI zu ändern. Weitere Informationen finden Sie `ModifyDBCluster`in der Amazon DocumentDB DocumentDB-API-Referenz.

Parameter

- **`--db-cluster-identifier`**—Erforderlich. Die ID des Amazon DocumentDB-Clusters, den Sie ändern werden.
- **`--backup-retention-period`**— Fakultativ. Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Gültige Werte sind 1—35.

- **`--storage-type`**— Fakultativ. Die Speicherkonfiguration des Clusters. Gültige Werte sind `standard` (Standard) oder `iopt1` (I/O-optimiert).
- **`--db-cluster-parameter-group-name`**— Optional. Der Name der Cluster-Parametergruppe, die für den Cluster verwendet werden soll.
- **`--manage-master-user-password`**—Fakultativ. Amazon DocumentDB generiert das Masterbenutzerkennwort und verwaltet es während seines gesamten Lebenszyklus in Secrets Manager.
- **`--rotate-master-user-password`**— Fakultativ. Secrets Manager generiert eine neue geheime Version für das bestehende Geheimnis. Die neue Version des Secrets enthält das neue primäre Benutzerkennwort. Amazon DocumentDB ändert das primäre Benutzerkennwort für den Cluster so, dass es mit dem Passwort für die neue geheime Version übereinstimmt.

  Sie müssen die **`--apply-immediately`** Option angeben, wenn Sie das primäre Passwort ändern.
- **`--master-user-password`**— Fakultativ. Das neue Passwort für den primären Datenbankbenutzer.

  Passwortbeschränkungen:

  - Die Länge beträgt [8—100] druckbare ASCII-Zeichen.
  - Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
    - **`/`** (Schrägstrich)
    - **`"`** (doppeltes Anführungszeichen)
    - **`@`** ('At'-Symbol)
- **`--new-db-cluster-identifier`**— Fakultativ. Die neue Cluster-ID für den Cluster beim Umbenennen eines Clusters. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

  Benennungseinschränkungen:

  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Amazon Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.

- **`--preferred-backup-window`**— Fakultativ. Der tägliche Zeitbereich, in dem automatisierte Backups erstellt werden, in Universal Coordinated Time (UTC).
  - Format: `hh24:mm-hh24:mm`
- **`--preferred-maintenance-window`**—Fakultativ. Der wöchentliche Zeitbereich, in dem Systemwartungen stattfinden können, in UTC.
  - Format: `ddd:hh24:mm-ddd:hh24:mm`
  - Gültige Tage: `Sun`, `Mon`, `Tue`, `Wed`, `Thu`, `Fri` und `Sat`.
- **`--deletion-protection`**oder **`--no-deletion-protection`** — Optional. Gibt an, ob der Löschschutz für diesen Cluster aktiviert werden soll. Der Löschschutz verhindert, dass ein Cluster versehentlich gelöscht wird, solange der Cluster-Löschschutz nicht deaktiviert wurde. Weitere Informationen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.
- **`--apply-immediately`**oder **`--no-apply-immediately`** — Verwenden Sie diese Option`--apply-immediately`, um die Änderung sofort vorzunehmen. Verwenden Sie `--no-apply-immediately`, um die Änderung während des nächsten Wartungsfensters Ihres Clusters vorzunehmen.

Example

Der folgende Code ändert die Aufbewahrungsfrist für das Backup für den Cluster `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --apply-immediately \
       --backup-retention-period 7
```

Für Windows:

```
aws docdb modify-db-cluster ^
       --db-cluster-identifier sample-cluster ^
       --apply-immediately ^
       --backup-retention-period 7
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
```

```
    "DBCluster": {
        "BackupRetentionPeriod": 7,
        "DbClusterResourceId": "cluster-VDP53QEWST7YHM36TTXOPJT5YE",
        "Status": "available",
        "DBClusterMembers": [
            {
                "PromotionTier": 1,
                "DBClusterParameterGroupStatus": "in-sync",
                "DBInstanceIdentifier": "sample-cluster-instance",
                "IsClusterWriter": true
            }
        ],
        "ReadReplicaIdentifiers": [],
        "AvailabilityZones": [
            "us-east-1b",
            "us-east-1c",
            "us-east-1a"
        ],
        "ReaderEndpoint": "sample-cluster.cluster-ro-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "PreferredMaintenanceWindow": "sat:09:51-sat:10:21",
        "EarliestRestorableTime": "2018-06-17T00:06:19.374Z",
        "StorageEncrypted": false,
        "MultiAZ": false,
        "AssociatedRoles": [],
        "MasterUsername": "<your-master-user-name>",
        "DBClusterIdentifier": "sample-cluster",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "HostedZoneId": "Z2SUY0A1719RZT",
        "LatestRestorableTime": "2018-06-18T21:17:05.737Z",
        "AllocatedStorage": 1,
        "Port": 27017,
        "Engine": "docdb",
        "DBClusterParameterGroup": "default.docdb3.4",
        "Endpoint": "sample-cluster.cluster-ctevjxdlur57.us-
east-1.rds.amazonaws.com",
        "DBSubnetGroup": "default",
        "PreferredBackupWindow": "00:00-00:30",
```

```
        "EngineVersion": "3.4",
        "ClusterCreateTime": "2018-06-06T19:25:47.991Z",
        "IAMDatabaseAuthenticationEnabled": false
    }
}
```

Es dauert einige Minuten, bis Ihre Änderungen übernommen werden. Sie können den Cluster nur verwenden, wenn sein Status available ist. Sie können den Status des Clusters über die Konsole oder AWS CLIüberwachen. Weitere Informationen finden Sie unter Den Status eines Amazon DocumentDB-Clusters überwachen.

Ermittlung ausstehender Wartungsarbeiten

Sie können feststellen, ob Sie über die neueste Version der Amazon DocumentDB DocumentDB-Engine verfügen, indem Sie feststellen, ob die Cluster-Wartung noch aussteht.

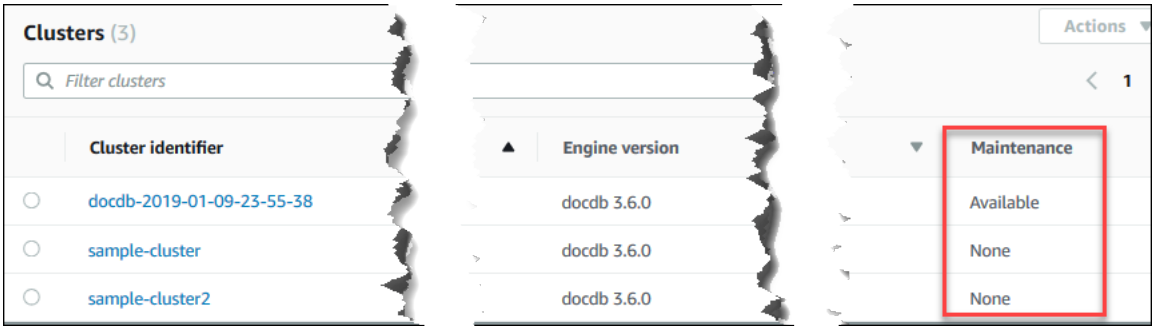
Using the AWS Management Console

Sie können den verwenden AWS Management Console , um festzustellen, ob für einen Cluster noch Wartungsarbeiten anstehen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Überprüfen Sie die Spalte Maintenance (Wartung), um festzustellen, ob ein Cluster eine ausstehende Wartung hat.



None (Keine) bedeutet, dass auf dem Cluster die neueste Engine-Version ausgeführt wird. Available (Verfügbar) bedeutet, dass der Cluster noch nicht gewartet wurde. Somit ist ein Engine-Upgrade erforderlich.

4. Wenn Ihr Cluster noch nicht gewartet wurde, fahren Sie mit den Schritten unter Durchführen eines Patch-Updates für die Engine-Version eines Clusters fort.

Using the AWS CLI

Sie können den verwenden AWS CLI , um festzustellen, ob ein Cluster über die neueste Engine-Version verfügt, indem Sie den `describe-pending-maintenance-actions` Vorgang mit den folgenden Parametern verwenden.

Parameter

- **`--resource-identifier`**— Optional. Der ARN der Ressource (Cluster). Wenn dieser Parameter weggelassen wird, werden ausstehende Wartungsaktionen für alle Cluster aufgelistet.
- **`--region`**—Fakultativ. Die AWS-Region, in der Sie diese Operation durchführen möchten (z. B. `us-east-1`).

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-actions \
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster \
   --region us-east-1
```

Für Windows:

```
aws docdb describe-pending-maintenance-actions ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --region us-east-1
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:123456789012:cluster:sample-cluster",
            "PendingMaintenanceActionDetails": [
                {
                    "Description": "New feature",
                    "Action": "db-upgrade",
                    "ForcedApplyDate": "2019-02-25T21:46:00Z",
                    "AutoAppliedAfterDate": "2019-02-25T07:41:00Z",
                    "CurrentApplyDate": "2019-02-25T07:41:00Z"
                }
            ]
        }
    ]
}
```

Wenn Ihr Cluster noch nicht gewartet wurde, fahren Sie mit den Schritten unter Durchführen eines Patch-Updates für die Engine-Version eines Clusters fort.

Durchführen eines Patch-Updates für die Engine-Version eines Clusters

In diesem Abschnitt erklären wir, wie Sie ein Patch-Update mit dem AWS Management Console oder dem bereitstellen AWS CLI. Ein Patch-Update ist ein Update innerhalb derselben Engine-Version (z. B. die Aktualisierung einer 3.6-Engine-Version auf eine neuere 3.6-Engine-Version). Sie können es sofort oder während des nächsten Wartungsfensters Ihres Clusters aktualisieren. Informationen darüber, ob Ihre Engine aktualisiert werden muss, finden Sie unterErmittlung ausstehender Wartungsarbeiten. Bitte beachten Sie, dass es bei der Installation des Updates zu einigen Ausfallzeiten auf Ihrem Cluster kommen kann.

> **Note**
>
> Wenn Sie versuchen, ein Upgrade von einer Hauptversion der Engine auf eine andere durchzuführen, z. B. 3.6 auf 5.0, finden Sie entweder Direktes Upgrade der Hauptversion von Amazon DocumentDB oderAktualisieren Ihres Amazon DocumentDB-Clusters mit AWS Database Migration Service. Ein direktes Upgrade der Hauptversion unterstützt nur docdb 5.0 als Ziel-Engine-Version.

Es gibt zwei Konfigurationsanforderungen, um die neuesten Patch-Updates für die Engine-Version eines Clusters zu erhalten:

- Der Cluster-Status muss available (verfügbar) lauten.
- Der Cluster muss eine ältere Engine-Version ausführen.

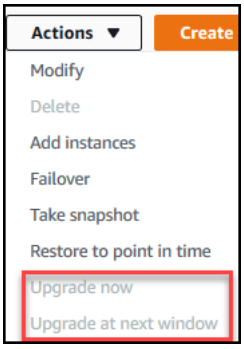
Using the AWS Management Console

Das folgende Verfahren wendet mithilfe der Konsole Patch-Updates auf die Engine-Version Ihres Clusters an. Sie haben die Möglichkeit, das Update sofort oder während des nächsten Wartungsfensters Ihres Clusters durchzuführen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Cluster aus, den Sie upgraden möchten. Der Cluster-Status muss available (verfügbar) lauten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Menü Actions (Aktionen) eine der folgenden Optionen aus. Diese Menüoptionen sind nur auswählbar, wenn der ausgewählte Cluster nicht die neueste Engine-Version ausführt.



- Jetzt aktualisieren — Startet sofort den Upgrade-Vorgang. Ihr Cluster ist eine zeitlang offline, während er auf die neueste Engine-Version aktualisiert wird.
- Upgrade im nächsten Fenster — Initiiert den Upgrade-Vorgang während des nächsten Wartungsfensters des Clusters. Ihr Cluster ist eine zeitlang offline, während er auf die neueste Engine-Version aktualisiert wird.

4. Wenn sich das Bestätigungsfenster öffnet, wählen Sie eine der folgenden Optionen aus:
   - Upgrade — Um Ihren Cluster gemäß dem im vorherigen Schritt ausgewählten Zeitplan auf die neueste Engine-Version zu aktualisieren.
   - Abbrechen — Um das Engine-Upgrade des Clusters abzubrechen und mit der aktuellen Engine-Version des Clusters fortzufahren.

Using the AWS CLI

Sie können Patch-Updates auf Ihren Cluster anwenden, indem Sie den `apply-pending-maintenance-action` Vorgang AWS CLI und mit den folgenden Parametern verwenden.

Parameter

- **`--resource-identifier`**—Erforderlich. Der ARN des Amazon DocumentDB-Clusters, den Sie aktualisieren möchten.
- **`--apply-action`**—Erforderlich. Die folgenden Werte sind zulässig. Verwenden Sie `db-upgrade`, um Ihre Cluster-Engine-Version upzugraden.
  - **`db-upgrade`**
  - **`system-update`**
- **`--opt-in-type`**—Erforderlich. Die folgenden Werte sind zulässig.
  - `immediate`— Wenden Sie die Wartungsmaßnahme sofort an.
  - `next-maintenance`— Wendet die Wartungsaktion im nächsten Wartungsfenster an.

- `undo-opt-in`— Stornieren Sie alle bestehenden `next-maintenance` Opt-In-Anfragen.

Example

Mit dem folgenden Beispiel-Patch wird die Engine-Version von `sample-cluster` auf Version 4.0.0 aktualisiert.

Für Linux, macOS oder Unix:

```
aws docdb apply-pending-maintenance-action \
   --resource-identifier arn:aws:rds:us-east-1:123456789012\:cluster:sample-cluster
 \
   --apply-action db-upgrade \
   --opt-in-type immediate
```

Für Windows:

```
aws docdb apply-pending-maintenance-action ^
   --resource-identifier arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster ^
   --apply-action db-upgrade ^
   --opt-in-type immediate
```

Die Ausgabe dieser Operation sieht folgendermaßen aus.

```
{
    "ResourcePendingMaintenanceActions": {
        "ResourceIdentifier": "arn:aws:rds:us-
east-1:444455556666:cluster:docdb-2019-01-09-23-55-38",
        "PendingMaintenanceActionDetails": [
            {
                "CurrentApplyDate": "2019-02-20T20:57:06.904Z",
                "Description": "Bug fixes",
                "ForcedApplyDate": "2019-02-25T21:46:00Z",
                "OptInStatus": "immediate",
                "Action": "db-upgrade",
                "AutoAppliedAfterDate": "2019-02-25T07:41:00Z"
            }
        ]
    }
}
```

Einen Amazon DocumentDB-Cluster stoppen und starten

Das Stoppen und Starten von Amazon DocumentDB-Clustern kann Ihnen helfen, die Kosten für Entwicklungs- und Testumgebungen zu verwalten. Anstatt jedes Mal, wenn Sie Amazon DocumentDB verwenden, Cluster und Instances zu erstellen und zu löschen, können Sie alle Instances in Ihrem Cluster vorübergehend beenden, wenn sie nicht benötigt werden. Sie können sie neu starten, wenn Sie Ihre Tests fortsetzen.

Themen

- Übersicht über das Stoppen und Starten eines Clusters
- Operationen, die Sie auf einem gestoppten Cluster ausführen können

Übersicht über das Stoppen und Starten eines Clusters

In Zeiten, in denen Sie keinen Amazon DocumentDB-Cluster benötigen, können Sie alle Instances in diesem Cluster gleichzeitig stoppen. Bei Bedarf können Sie den Cluster jederzeit erneut starten. Durch das Starten und Stoppen werden die Einrichtungs- und Entfernungsvorgänge für Cluster erleichtert, die in der Entwicklung, für Tests oder ähnliche Aktivitäten verwendet werden und keine kontinuierliche Verfügbarkeit erfordern. Sie können einen Cluster mithilfe von AWS Management Console oder AWS CLI mit einer einzigen Aktion beenden und starten, unabhängig davon, wie viele Instances sich im Cluster befinden.

Während Ihr Cluster gestoppt ist, bleibt des Cluster-Speichervolumen unverändert. Sie zahlen nur für Speicherung, manuelle Snapshots und automatischen Sicherungsspeicher innerhalb des angegebenen Aufbewahrungsfensters. Ihnen werden keine Instance-Stunden in Rechnung gestellt. Amazon DocumentDB startet Ihren Cluster automatisch nach sieben Tagen, sodass er nicht hinter den erforderlichen Wartungsupdates zurückbleibt. Wenn Ihr Cluster nach sieben Tagen wieder startet, werden Ihnen die Instances im Cluster wieder in Rechnung gestellt. Während Ihr Cluster gestoppt ist, können Sie Ihr Speichervolumen nicht abfragen, da die Abfrage erfordert, dass sich Instances im Status „available (verfügbar)“ befinden.

Wenn ein Amazon DocumentDB-Cluster gestoppt wird, können weder der Cluster noch seine Instances in irgendeiner Weise geändert werden. Dies umfasst das Hinzufügen oder Entfernen von Instances oder das Löschen des Clusters.

Using the AWS Management Console

Das folgende Verfahren zeigt, wie Sie einen Cluster mit einer oder mehreren Instances im Status „verfügbar“ stoppen oder einen gestoppten Cluster wieder starten.

So beenden oder starten Sie einen Amazon DocumentDB-Cluster

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

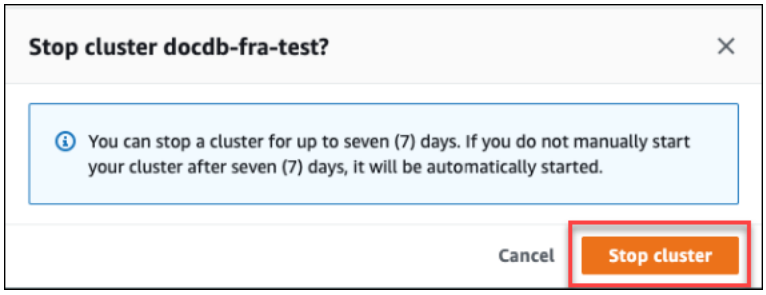
> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Namen des Clusters aus, den Sie stoppen oder starten möchten.
4. Wählen Sie Aktionen aus und dann die Aktion, die Sie auf dem Cluster ausführen möchten.
   - Wenn Sie den Cluster stoppen möchten und der Cluster verfügbar ist:
     a. Wählen Sie Beenden aus.

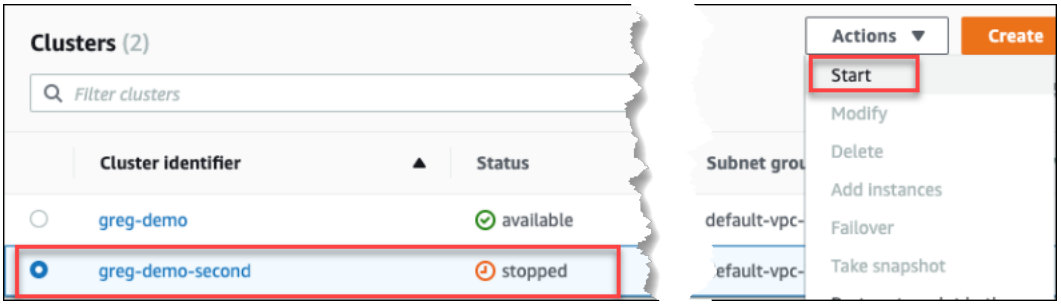


Damit der Failover-Mechanismus nicht aktiviert wird, werden beim Stoppvorgang zuerst die Replikat-Instances und dann die primäre Instance gestoppt.

     b. Wenn Sie den Cluster stoppen möchten, wählen Sie im Bestätigungsdialogfeld Cluster stoppen aus und wenn der Cluster weiter ausgeführt werden soll, wählen Sie Abbrechen aus.



- Wenn Sie einen gestoppten Cluster starten möchten, wählen Sie Starten aus.



5. Überwachen Sie den Status des Clusters und seiner Instances. Wenn Sie den Cluster gestartet haben, können Sie die Ausführung des Clusters fortsetzen, wenn der Cluster und dessen Instances verfügbar sind. Weitere Informationen finden Sie unter Den Status eines Clusters ermitteln.



Using the AWS CLI

Die folgenden Codebeispiele zeigen, wie Sie einen Cluster mit einer oder mehreren Instances im Status „verfügbar“ stoppen oder einen gestoppten Cluster starten.

Um einen Cluster mit einer oder mehreren verfügbaren Instances mithilfe von zu beenden AWS CLI, verwenden Sie den Vorgang. `stop-db-cluster` Verwenden Sie die `start-db-cluster`-Operation, um einen gestoppten Cluster zu starten. Beide Operationen verwenden den `--db-cluster-identifier`-Parameter.

Parameter:

- **`--db-cluster-identifier`**—Erforderlich. Der Name des Clusters, der gestoppt oder gestartet werden soll.

Example — Um einen Cluster mit dem zu beenden AWS CLI

Der folgende Code stoppt den Cluster `sample-cluster`. Der Cluster muss eine oder mehrere Instances im verfügbaren Zustand haben.

Für Linux, macOS oder Unix:

```
aws docdb stop-db-cluster \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb stop-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Example — Um einen Cluster mit dem zu starten AWS CLI

Der folgende Code startet den Cluster `sample-cluster`. Der Cluster muss jetzt gestoppt werden.

Für Linux, macOS oder Unix:

```
aws docdb start-db-cluster \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb start-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Operationen, die Sie auf einem gestoppten Cluster ausführen können

Während ein Amazon DocumentDB-Cluster gestoppt ist, können Sie eine point-in-time Wiederherstellung an einem beliebigen Punkt innerhalb des angegebenen Zeitfensters für die

automatische Aufbewahrung von Backups durchführen. Einzelheiten zur Durchführung einer point-in-time Wiederherstellung finden Sie unterZu einem bestimmten Zeitpunkt wiederherstellen.

Sie können die Konfiguration eines Amazon DocumentDB-Clusters oder einer seiner Instances nicht ändern, solange der Cluster gestoppt ist. Es ist auch nicht möglich, Instances zum Cluster hinzuzufügen oder daraus zu entfernen oder den Cluster zu löschen, wenn ihm noch Instances zugeordnet sind. Vor solchen administrativen Aktionen müssen Sie den Cluster starten.

Amazon DocumentDB wendet alle geplanten Wartungsarbeiten auf Ihren gestoppten Cluster erst an, nachdem er erneut gestartet wurde. Nach sieben Tagen startet Amazon DocumentDB automatisch einen gestoppten Cluster, sodass er in seinem Wartungsstatus nicht zu weit zurückfällt. Sobald Ihr Cluster neu gestartet wird, werden Ihnen die Instances im Cluster wieder in Rechnung gestellt.

Solange ein Cluster gestoppt ist, führt Amazon DocumentDB keine automatisierten Backups durch und verlängert auch nicht den Aufbewahrungszeitraum für Backups.

Löschen eines Amazon DocumentDB-Clusters

Sie können einen Amazon DocumentDB-Cluster mit dem AWS Management Console oder dem AWS CLI löschen. Um einen Cluster zu löschen, muss sich der Cluster im Status available (verfügbar) befinden und es dürfen keine Instances mit ihm verknüpft sein. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis der Cluster verfügbar wird, und löschen Sie dann den Cluster. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster stoppen und starten.

Löschschutz

Zum Schutz Ihrer Cluster vor versehentlichem Löschen, können Sie den Löschschutz aktivieren. Wenn Sie mithilfe der Konsole einen Cluster erstellen, ist der Löschschutz standardmäßig aktiviert. Der Löschschutz ist jedoch standardmäßig deaktiviert, wenn Sie einen Cluster über die AWS CLI erstellen.

Amazon DocumentDB erzwingt den Löschschutz für einen Cluster, unabhängig davon, ob Sie den Löschvorgang über die Konsole oder die ausführen. AWS CLI Sie können einen Cluster nicht löschen, solange der Löschschutz aktiviert ist. Um einen Cluster mit aktiviertem Löschschutz zu löschen, müssen Sie zuerst den Cluster ändern und den Löschschutz deaktivieren.

Wenn Sie die Konsole mit aktiviertem Löschschutz für einen Cluster verwenden, können Sie die letzte Instance des Clusters nicht löschen, da dadurch auch der Cluster gelöscht werden würde. Sie können die letzte Instance eines mit Löschschutz geschützten Clusters mit der AWS CLI löschen. In diesen

Fall bleibt der Cluster selbst bestehen und die Daten werden beibehalten. Sie können zum Zugriff auf die Daten neue Instances für den Cluster erstellen. Weitere Informationen zum Aktivieren und Deaktivieren des Löschschutzes siehe:

- Einen Amazon DocumentDB-Cluster erstellen
- Ändern eines Amazon DocumentDB-Clusters

Using the AWS Management Console

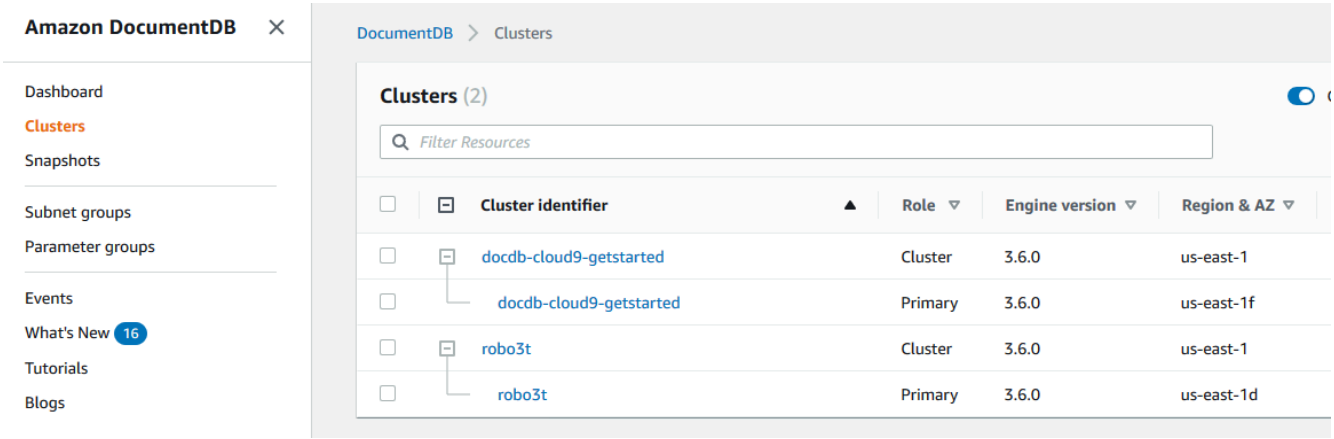
Um einen Cluster mit dem zu löschen AWS Management Console, muss der Löschschutz deaktiviert sein.

So finden Sie heraus, ob der Löschschutz für einen Cluster aktiviert ist:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Beachten Sie, dass im Cluster-Navigationsfeld in der Spalte Cluster-ID sowohl Cluster als auch Instances angezeigt werden. Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Wählen Sie den Namen des Clusters und die Registerkarte Configuration (Konfiguration) aus. Suchen Sie im Abschnitt Cluster-Details die Option Deletion protection (Löschschutz). Wenn der Löschschutz aktiviert ist, bearbeiten Sie den Cluster, um den Löschschutz zu deaktivieren. Informationen über das Ändern eines Clusters finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

   Nachdem Sie den Löschschutz deaktiviert haben, können Sie den Cluster löschen.

So löschen Sie einen Cluster:

1. Klicken Sie im Navigationsbereich auf Cluster.
2. Ermitteln Sie, ob der Cluster über Instances verfügt, indem Sie in der Spalte Cluster-ID nach den darunter aufgelisteten Instances suchen. Bevor Sie einen Cluster löschen können, müssen Sie alle seine Instances löschen. Weitere Informationen finden Sie unter Löschen einer Amazon DocumentDB DocumentDB-Instance.
3. Abhängig davon, ob Ihr Cluster Instances hat oder nicht, führen Sie einen der folgenden Schritte aus:
   - Wenn der Cluster über keine Instances verfügt, wählen Sie die Schaltfläche links neben dem Cluster-Namen und dann Actions (Aktionen) aus. Wählen Sie im Dropdown-Menü Löschen aus. Füllen Sie das Dialogfeld <cluster-name> löschen aus und klicken Sie dann auf Löschen.
   - Wenn der Cluster eine oder mehrere Instances hat, gehen Sie wie folgt vor:
     a. Klicken Sie im Navigationsbereich auf Cluster.
     b. Löschen Sie alle Instanzen des Clusters, indem Sie das Kontrollkästchen links neben dem Namen des Clusters aktivieren. Wählen Sie Actions (Aktionen) und anschließend Delete (Löschen) aus. Füllen Sie das Dialogfeld <cluster-name> löschen aus und klicken Sie dann auf Löschen.

        Wenn Sie die letzte Instanz löschen, wird auch der Cluster gelöscht. Weitere Informationen zum Löschen von Instanzen finden Sie unterLöschen einer Amazon DocumentDB DocumentDB-Instance.

Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Status des Clusters finden Sie unter Den Status eines Amazon DocumentDB-Clusters überwachen.

Using the AWS CLI

Sie können einen Cluster, dem Instances zugeordnet sind, nicht löschen. Um festzustellen, welche Instances mit dem Cluster verknüpft sind, führen Sie den Befehl `describe-db-clusters` aus und löschen Sie alle Instances des Clusters. Deaktivieren Sie dann bei Bedarf den Löschschutz auf Ihrem Cluster und löschen Sie schließlich den Cluster.

1. Löschen Sie zuerst alle Instances des Clusters.

   Um festzustellen, welche Instances Sie löschen müssen, führen Sie den folgenden Befehl aus.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].
   [DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
   ```

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

   ```
   [
       [
           "sample-cluster",
           [
               "sample-instance-1",
               "sample-instance-2"
           ]
       ]
   ]
   ```

   Wenn der Cluster, den Sie löschen möchten, Instances enthält, löschen Sie diese wie unten dargestellt.

   ```
   aws docdb delete-db-instance \
       --db-instance-identifier sample-instance
   ```

2. Zweitens deaktivieren Sie den Löschschutz.

   Wenn Sie die AWS CLI Option verwenden, um alle Instanzen eines Clusters zu löschen, wird der Cluster nicht gelöscht. Sie müssen auch den Cluster löschen, aber dies ist nur möglich, wenn der Löschschutz deaktiviert ist.

Um festzustellen, ob der Löschschutz für den Cluster aktiviert ist, führen Sie den folgenden Befehl aus.

> **Tip**
>
> Um den Löschschutzstatus all Ihrer Amazon DocumentDB-Cluster zu sehen, lassen Sie den `--db-cluster-identifier` Parameter weg.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterIdentifier,DeletionProtection]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-cluster",
        "true"
    ]
]
```

Wenn für einen Cluster der Löschschutz aktiviert ist, bearbeiten Sie den Cluster, um den Löschschutz zu deaktivieren. Führen Sie den folgenden Befehl aus, um den Löschschutz des Clusters zu deaktivieren.

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --no-deletion-protection \
    --apply-immediately
```

3. Löschen Sie schließlich den Cluster.

   Nachdem Sie den Löschschutz deaktiviert haben, können Sie den Cluster löschen. Um einen Cluster zu löschen, verwenden Sie die `delete-db-cluster`-Operation mit den folgenden Parametern.

- **`--db-cluster-identifier`**—Erforderlich. Die ID des Clusters, den Sie löschen möchten.
- **`--final-db-snapshot-identifier`**— Fakultativ. Wenn Sie einen endgültigen Snapshot wünschen, müssen Sie diesen Parameter mit einem Namen für den endgültigen Snapshot angeben. Sie müssen entweder `--final-db-snapshot-identifier` oder `--skip-final-snapshot` angeben.

  Benennungseinschränkungen:

  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Cluster in Amazon RDS, Amazon Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.
- **`--skip-final-snapshot`**— Fakultativ. Verwenden Sie diesen Parameter nur, wenn Sie vor dem Löschen Ihres Clusters keinen letzten Snapshot machen möchten. Standardmäßig wird ein letzter Snapshot erstellt. Sie müssen entweder `--final-db-snapshot-identifier` oder `--skip-final-snapshot` angeben.

Der folgende AWS CLI Code löscht den Cluster `sample-cluster` mit einem endgültigen Snapshot. Die Operation schlägt fehl, wenn Instances vorhanden sind, die mit dem Cluster verknüpft sind, oder wenn der Löschschutz aktiviert ist.

Example

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Für Windows:

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --final-db-snapshot-identifier sample-cluster-final-snapshot
```

Example

Der folgende AWS CLI Code löscht den Cluster, `sample-cluster` ohne einen endgültigen Snapshot zu erstellen.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

Für Windows:

```
aws docdb delete-db-cluster ^
    --db-cluster-identifier sample-cluster ^
    --skip-final-snapshot
```

Die Ausgabe der `delete-db-cluster`-Operation ist der Cluster, den Sie löschen.

Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Status des Clusters finden Sie unter Den Status eines Clusters überwachen.

Skalierung von Amazon DocumentDB-Clustern

Mit Amazon DocumentDB können Sie den Speicher und die Rechenleistung in Ihren Clustern nach Ihren Bedürfnissen skalieren. In diesem Abschnitt wird beschrieben, wie Sie Speicherskalierung, Instanzskalierung und Leseskalierung verwenden können, um die Leistung und Skalierung für Ihre Amazon DocumentDB-Cluster und -Instances zu verwalten.

Themen

- Speicherskalierung
- Skalierung von Instances
- Skalierung von Lesevorgängen
- Skalierung schreiben

Speicherskalierung

Der Amazon DocumentDB DocumentDB-Speicher wird automatisch mit den Daten in Ihrem Cluster-Volume skaliert. Wenn Ihre Daten wachsen, wächst Ihr Cluster-Volumenspeicher in Schritten von 10 GiB auf bis zu 128 TiB.

Skalierung von Instances

Sie können Ihren Amazon DocumentDB-Cluster nach Bedarf skalieren, indem Sie die Instance-Klasse für jede Instance im Cluster ändern. Amazon DocumentDB unterstützt mehrere Instance-Klassen, die für Amazon DocumentDB optimiert sind.

Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB DocumentDB-Instance.

Skalierung von Lesevorgängen

Sie können die Leseskalierung für Ihren Amazon DocumentDB-Cluster erreichen, indem Sie bis zu 15 Amazon DocumentDB DocumentDB-Replikate im Cluster erstellen. Jedes Amazon DocumentDB DocumentDB-Replikat gibt dieselben Daten aus dem Cluster-Volume mit minimaler Replikatverzögerung zurück — normalerweise weniger als 100 Millisekunden, nachdem die primäre Instance ein Update geschrieben hat. Wenn Ihr Lesedatenverkehr zunimmt, können Sie zusätzliche Amazon DocumentDB DocumentDB-Replikate erstellen und sich direkt mit ihnen verbinden, um die Leselast für Ihren Cluster zu verteilen. Amazon DocumentDB DocumentDB-Replikate müssen nicht derselben Instance-Klasse angehören wie die primäre Instance.

Weitere Informationen finden Sie unter Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster.

Um den Lesesatz mit Amazon DocumentDB zu nutzen, empfehlen wir, dass Sie eine Verbindung zu Ihrem Cluster als Replikatsatz herstellen und Lesevorgänge mithilfe der integrierten Leseeinstellungen Ihres Treibers an Replikatinstanzen verteilen. Weitere Informationen finden Sie unter Als Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen.

Skalierung schreiben

Sie können die Schreibkapazität in Ihrem Amazon DocumentDB-Cluster skalieren, indem Sie die Größe der primären Instance Ihres Clusters erhöhen. In diesem Abschnitt werden zwei Methoden für die Skalierung der primären Instance Ihres Clusters abhängig von Ihren Anforderungen beschrieben. Die erste Option zielt darauf ab, die Auswirkungen auf die Anwendung so gering wie möglich

zu halten, erfordert jedoch mehr Schritte. Die zweite Option ermöglicht Ihnen die Optimierung in Bezug auf die Einfachheit, da sie weniger Schritte erfordert. Die potenziellen Auswirkungen auf Anwendungen sind jedoch größer.

Abhängig von Ihrer Anwendung können Sie wählen, welcher Ansatz für Sie am besten geeignet ist. Weitere Informationen zu verfügbaren Instance-Größen und Kosten finden Sie auf der Seite mit den Amazon DocumentDB DocumentDB-Preisen.

1. Optimieren Sie für hohe Verfügbarkeit und Leistung — Wenn Sie im Replica-Set-Modus (empfohlen) eine Verbindung zu Ihrem Cluster herstellen, können Sie den folgenden Prozess verwenden, um die Auswirkungen auf Ihre Anwendung bei der Skalierung Ihrer primären Instance zu minimieren. Diese Methode minimiert die Auswirkungen, da Ihr Cluster den von Ihnen festgelegten Wert für Hochverfügbarkeit weiter einhält oder überschreitet. Darüber hinaus werden Leseskalierungsziele dem Cluster als Instances hinzugefügt, anstatt an Ort und Stelle aktualisiert zu werden.

   a. Fügen Sie ein oder mehrere Replikate des größeren Instance-Typs zu Ihrem Cluster hinzu (siehe ???). Alle Replikate sollten einen Instance-Typ besitzen, der größer oder gleich dem Typ der primären Instance ist. Dies vermeidet eine unbeabsichtigte Reduzierung der Schreibleistung aufgrund eines Failovers zu einem kleineren Instance-Typ. Für die meisten Kunden bedeutet dies, dass die Anzahl der Instances in ihrem Cluster vorübergehend verdoppelt wird und die kleineren Replikate nach Abschluss der Skalierung entfernt werden.

   b. Legen Sie die Failover-Stufe für alle neuen Replikate auf die Priorität Null fest. Dies stellt sicher, dass ein Replikat des kleineren Instance-Typs die höchste Failover-Priorität besitzt. Weitere Informationen finden Sie unter ???.

   c. Initiieren Sie einen manuellen Failover. Hierdurch wird eines der neuen Replikate zur primären Instance hochgestuft. Weitere Informationen finden Sie unter ???.

      > **Note**
      >
      > Dies führt zu einer Ausfallzeit von ca. 30 Sekunden für Ihren Cluster. Bitte planen Sie entsprechend.

   d. Entfernen Sie aus dem Cluster alle Replikate von Instance-Typen, die kleiner als der neue primäre Instance-Typ sind.

   e. Setzen Sie die Failover-Stufe aller Instances auf dieselbe Priorität zurück. (In der Regel bedeutet dies, dass sie auf 1 zurückgesetzt werden.)

Angenommen, Sie führen einen Cluster aus, der zurzeit drei `r5.large`-Instances enthält (eine primäre Instance und zwei Replikate). Sie möchten diese Instances auf den Instance-Typ `r5.xlarge` skalieren. Hierzu fügen Sie dem Cluster zunächst drei `r5.xlarge`-Replikat-Instances hinzu und legen anschließend die Failover-Stufe der neuen `r5.xlarge`-Replikate auf null fest. Als Nächstes leiten Sie einen manuellen Failover ein. (Ihnen ist bekannt, dass dies zu einer Ausfallzeit von ca. 30 Sekunden für Ihre Anwendung führt.) Nach Abschluss des Failovers entfernen Sie alle drei `r5.large`-Instances aus dem Cluster. Der Cluster ist nun auf `r5.xlarge`-Instances skaliert.

Um die Kosten zu optimieren, werden Amazon DocumentDB DocumentDB-Instances in Sekundenschritten abgerechnet, wobei eine Mindestgebühr von zehn Minuten nach einer Änderung des fakturierbaren Status, z. B. beim Erstellen, Ändern oder Löschen einer Instance, anfällt. Weitere Informationen finden Sie unter Kostenoptimierung in der Dokumentation zu bewährten Methoden.

2. Aus Gründen der Einfachheit optimieren — Dieser Ansatz ist aus Gründen der Einfachheit optimiert. Dadurch wird der Cluster nicht erweitert und verkleinert, aber es kann Ihre Lesekapazität vorübergehend reduzieren.

   Es ist möglich, dass das Ändern der Instanzklasse eines Replikats dazu führt, dass diese Instanz für einen kurzen Zeitraum, von einigen Sekunden bis weniger als 30 Sekunden, keine Anfragen bearbeitet. Wenn Sie im Replikat-Set-Modus (empfohlen) eine Verbindung zu Ihrem Cluster herstellen, würde dies Ihre Lesekapazität während des Skalierungsvorgangs um ein Replikat reduzieren (z. B. auf 66% Kapazität in einem 3-Knoten-Cluster oder 75% Kapazität in einem 4-Knoten-Cluster usw.).

   a. Skalieren Sie eine der Replikatinstanzen in Ihrem Cluster. Weitere Informationen finden Sie unter Instanzklassen verwalten.

   b. Warten Sie, bis die Instanz verfügbar ist (siheDen Status einer Amazon DocumentDB DocumentDB-Instance überwachen).

      > **Note**
      >
      > Dies führt zu einer Ausfallzeit von ca. 30 Sekunden für Ihren Cluster. Bitte planen Sie entsprechend.

c. Fahren Sie mit der Ausführung der Schritte 1 und 2 fort, bis alle Replikatinstanzen nacheinander skaliert wurden.

d. Initiieren Sie ein manuelles Failover. Dadurch wird eines der Replikate zur primären Instanz heraufgestuft. Weitere Informationen finden Sie unter Amazon DocumentDB-Failover.

> **Note**
>
> Dies führt zu einer Ausfallzeit von bis zu 30 Sekunden für Ihren Cluster, dauert aber oft weniger Zeit. Bitte planen Sie entsprechend.

e. Skalieren Sie die frühere primäre Instanz (jetzt ein Replikat).

Klonen eines Volumes für einen Amazon DocumentDB-Cluster

Durch das Klonen von Amazon DocumentDB können Sie einen neuen Cluster erstellen, der dasselbe Amazon DocumentDB-Cluster-Volume verwendet und dieselben Daten wie das Original hat. Der Prozess ist so konzipiert, dass er schnell und kostengünstig ist. Der neue Cluster mit dem zugehörigen Datenvolume wird als clone (Klon) bezeichnet. Das Erstellen eines Klons ist schneller und platzsparender als das physische Kopieren der Daten mit anderen Techniken, wie z. B. das Wiederherstellen eines Snapshots.

Amazon DocumentDB unterstützt die Erstellung eines von Amazon DocumentDB bereitgestellten Klons aus einem bereitgestellten Amazon DocumentDB-Cluster. Wenn Sie einen Clone mit einer anderen Bereitstellungskonfiguration als der Quelle erstellen, wird der Clone mit der neuesten Version der Amazon DocumentDB DocumentDB-Engine der Quelle erstellt.

Wenn Sie Klone aus Ihren Amazon DocumentDB-Clustern erstellen, werden die Klone in Ihrem Konto erstellt AWS — demselben Konto, dem der Amazon DocumentDB DocumentDB-Quellcluster gehört.

Themen

- Überblick über das Klonen von Amazon DocumentDB
- Einschränkungen beim Klonen von Amazon DocumentDB
- So funktioniert das Klonen von Amazon DocumentDB
- Einen Amazon DocumentDB-Klon erstellen

Überblick über das Klonen von Amazon DocumentDB

Amazon DocumentDB verwendet ein copy-on-write Protokoll, um einen Klon zu erstellen. Dieser Mechanismus verwendet minimalen zusätzlichen Speicherplatz, um einen ersten Klon zu erstellen. Wenn der Klon zum ersten Mal erstellt wird, behält Amazon DocumentDB eine einzige Kopie der Daten, die vom Quell-DB-Cluster und dem neuen (geklonten) Amazon DocumentDB-Cluster verwendet werden. Zusätzlicher Speicherplatz wird nur zugewiesen, wenn Änderungen an Daten (auf dem Amazon DocumentDB-Speichervolume) durch den Amazon DocumentDB-Quellcluster oder den Amazon DocumentDB-Cluster-Klon vorgenommen werden. Weitere Informationen über das copy-on-write Protokoll finden Sie unter. So funktioniert das Klonen von Amazon DocumentDB

Das Klonen von Amazon DocumentDB ist besonders nützlich, um schnell Testumgebungen mit Ihren Produktionsdaten einzurichten, ohne das Risiko einer Datenbeschädigung einzugehen. Sie können Klone für viele Arten von Anwendungen verwenden, z. B. für Folgende:

- Experimentieren Sie mit möglichen Änderungen (z. B. Schemaänderungen und Parametergruppenänderungen), um alle Auswirkungen zu bewerten.
- Führen Sie Workload-intensive Vorgänge aus, z. B. das Exportieren von Daten oder das Ausführen analytischer Abfragen auf dem Klon.
- Erstellen Sie eine Kopie Ihres Produktions-DB-Clusters zu Entwicklungs-, Test- oder anderen Zwecken.

Sie können mehr als einen Klon aus demselben Amazon DocumentDB-Cluster erstellen. Sie können auch mehrere Klone aus einem anderen Klon erstellen.

Nachdem Sie einen Amazon DocumentDB-Klon erstellt haben, können Sie die Amazon DocumentDB DocumentDB-Instances anders konfigurieren als der Amazon DocumentDB DocumentDB-Quellcluster. Beispielsweise benötigen Sie möglicherweise keinen Klon für Entwicklungszwecke, um dieselben Hochverfügbarkeitsanforderungen wie der Amazon DocumentDB DocumentDB-Quellproduktionscluster zu erfüllen. In diesem Fall können Sie den Clone mit einer einzigen Amazon DocumentDB-Instance statt mit mehreren DB-Instances konfigurieren, die vom Amazon DocumentDB-Cluster verwendet werden.

Wenn Sie den Klon für Test-, Entwicklungs- oder andere Zwecke nicht mehr verwenden, können Sie ihn löschen.

Einschränkungen beim Klonen von Amazon DocumentDB

Amazon DocumentDB; für das Klonen gelten derzeit die folgenden Einschränkungen:

- Sie können so viele Klone erstellen, wie Sie möchten, bis zur maximalen Anzahl von DB-Clustern, die in der AWS-Region zulässig sind. Nachdem Sie jedoch 15 Klone erstellt haben, ist der nächste Klon eine vollständige Kopie. Der Klonvorgang funktioniert wie eine Wiederherstellung. point-in-time
- Sie können keinen Clone in einer anderen AWS Region als dem Amazon DocumentDB DocumentDB-Quellcluster erstellen.
- Sie können keinen Clone aus einem Amazon DocumentDB-Cluster erstellen, der keine DB-Instances hat. Sie können nur Amazon DocumentDB-Cluster klonen, die über mindestens eine DB-Instance verfügen.
- Sie können einen Clone in einer anderen Virtual Private Cloud (VPC) als der des Amazon DocumentDB-Clusters erstellen. Wenn Sie dies tun, VPCs müssen die Subnetze von denselben Availability Zones zugeordnet werden.

So funktioniert das Klonen von Amazon DocumentDB

Das Klonen von Amazon DocumentDB funktioniert auf der Speicherebene eines Amazon DocumentDB-Clusters. Es verwendet ein copy-on-writeProtokoll, das sowohl schnell als auch platzsparend ist, was die zugrunde liegenden dauerhaften Medien angeht, die das Amazon DocumentDB DocumentDB-Speichervolumen unterstützen. Weitere Informationen zu Amazon DocumentDB-Cluster-Volumes finden Sie unterVerwaltung von Amazon DocumentDB-Clustern.

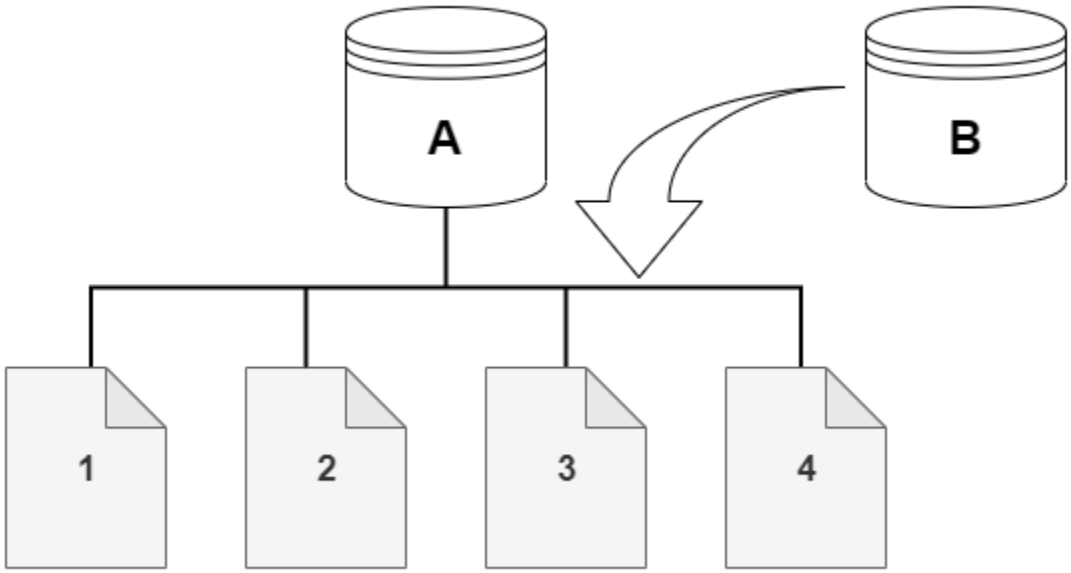
Themen

- Das Protokoll verstehen copy-on-write
- Löschen eines Quell-Cluster-Volumes

Das Protokoll verstehen copy-on-write

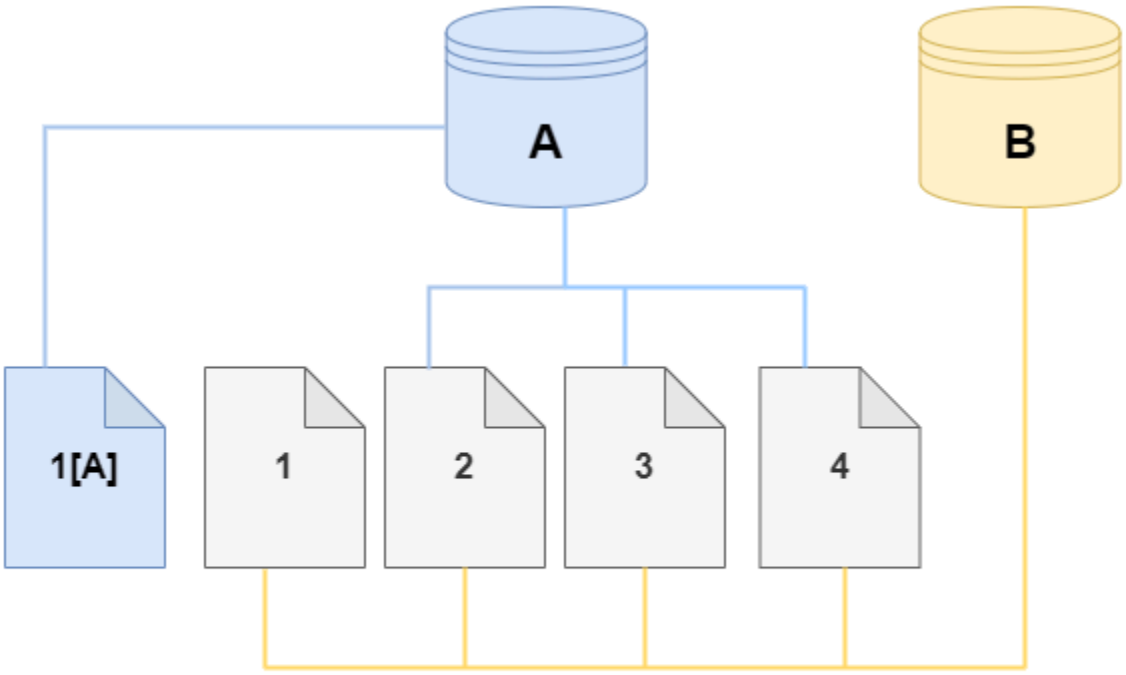
Ein Amazon DocumentDB-Cluster speichert Daten in Seiten auf dem zugrunde liegenden Amazon DocumentDB DocumentDB-Speichervolume.

Im folgenden Diagramm finden Sie beispielsweise einen Amazon DocumentDB-Cluster (A) mit vier Datenseiten, 1, 2, 3 und 4. Stellen Sie sich vor, dass ein Klon, B, aus dem Amazon DocumentDB-Cluster erstellt wird. Wenn der Klon erstellt wird, werden keine Daten kopiert. Vielmehr verweist der Klon auf denselben Seitensatz wie der Amazon DocumentDB DocumentDB-Quellcluster.

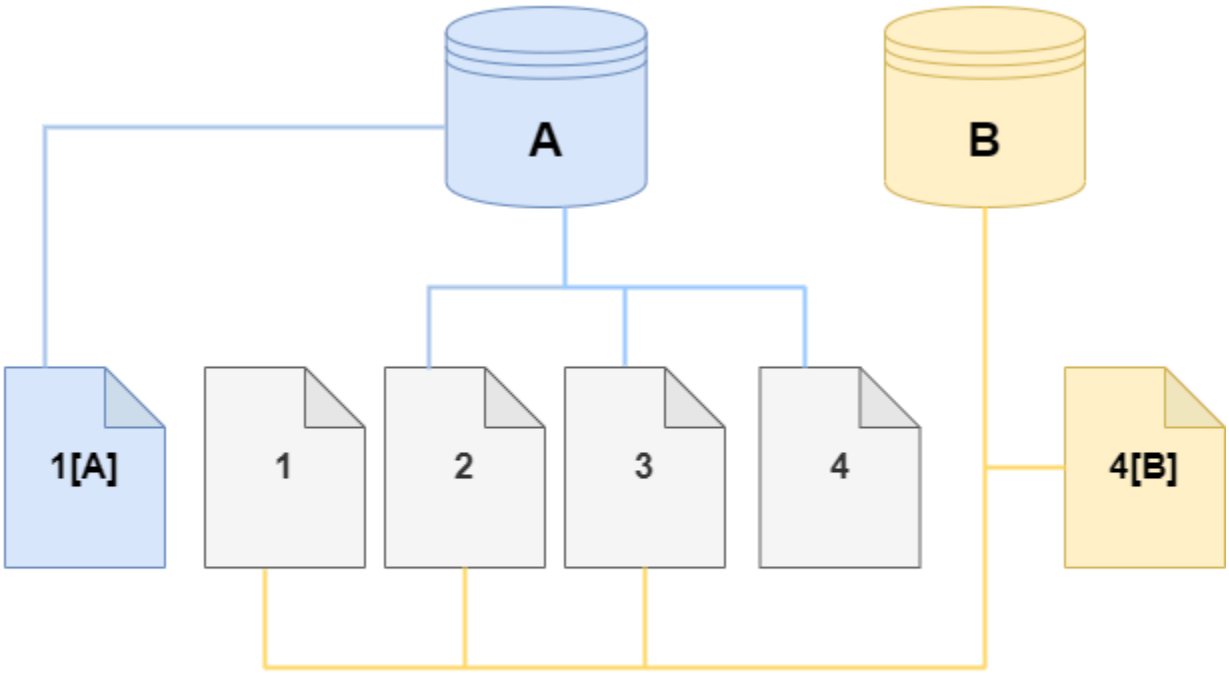


Wenn der Klon erstellt wird, ist normalerweise kein zusätzlicher Speicher erforderlich. Das copy-on-write Protokoll verwendet dasselbe Segment auf dem physischen Speichermedium wie das Quellsegment. Zusätzlicher Speicher ist nur erforderlich, wenn die Kapazität des Quellsegments für das gesamte Klonsegment nicht ausreicht. Wenn dies der Fall ist, wird das Quellsegment auf ein anderes physisches Gerät kopiert.

In den folgenden Diagrammen finden Sie ein Beispiel für das copy-on-write Protokoll in Aktion, das denselben Cluster A und seinen Klon B verwendet, wie oben gezeigt. Nehmen wir an, Sie nehmen eine Änderung an Ihrem Amazon DocumentDB-Cluster (A) vor, die zu einer Änderung der auf Seite 1 gespeicherten Daten führt. Anstatt auf die ursprüngliche Seite 1 zu schreiben, erstellt Amazon DocumentDB eine neue Seite 1 [A]. Das Amazon DocumentDB-Cluster-Volume für Cluster (A) verweist jetzt auf die Seiten 1 [A], 2, 3 und 4, während der Klon (B) immer noch auf die Originalseiten verweist.



Auf dem Klon wird eine Änderung an Seite 4 auf dem Speichervolume vorgenommen. Anstatt auf die ursprüngliche Seite 4 zu schreiben, erstellt Amazon DocumentDB eine neue Seite, 4 [B]. Der Klon verweist nun auf die Seiten 1, 2, 3 und auf Seite 4[B], während der Cluster (A) weiterhin auf 1[A], 2, 3 und 4 verweist.



Da im Laufe der Zeit immer mehr Änderungen sowohl im Amazon DocumentDB DocumentDB-Quell-Cluster-Volume als auch im Clone vorgenommen werden, wird mehr Speicherplatz benötigt, um die Änderungen zu erfassen und zu speichern.

Löschen eines Quell-Cluster-Volumes

Wenn Sie ein Quell-Cluster-Volume löschen, dem ein oder mehrere Klone zugeordnet sind, sind die Klone nicht betroffen. Die Klone verweisen weiter auf die Seiten, die zuvor im Besitz des Quell-Cluster-Volumes waren.

Einen Amazon DocumentDB-Klon erstellen

Sie können einen Clone in demselben AWS Konto wie der Amazon DocumentDB DocumentDB-Quellcluster erstellen. Dazu können Sie das AWS Management Console oder das AWS CLI und die folgenden Verfahren verwenden.

Mithilfe von Amazon DocumentDB-Cloning können Sie einen bereitgestellten Amazon DocumentDB-Cluster-Klon aus einem bereitgestellten Amazon DocumentDB-Cluster erstellen.

Using the AWS Management Console

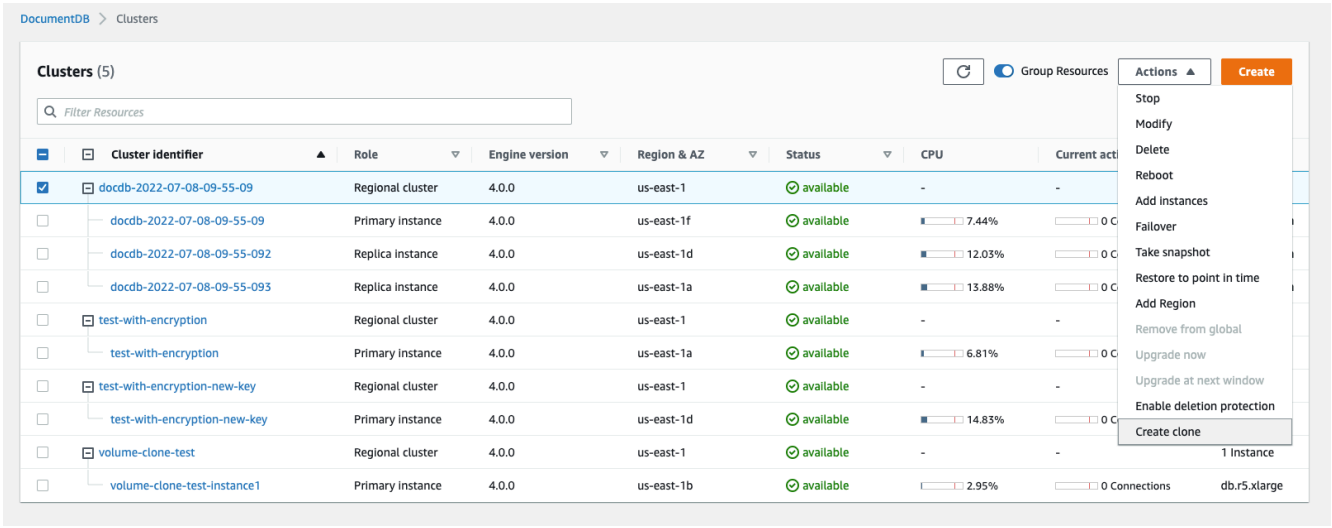
Das folgende Verfahren beschreibt, wie Sie einen Amazon DocumentDB-Cluster mit dem AWS Management Console klonen.

Erstellen eines Klons anhand der AWS Management Console Ergebnisse in einem Amazon DocumentDB-Cluster mit einer Amazon DocumentDB DocumentDB-Instance.

Diese Anweisungen gelten für DB-Cluster, die demselben AWS Konto gehören, das den Klon erstellt. Der DB-Cluster muss demselben AWS Konto gehören, da kontoübergreifendes Klonen in Amazon DocumentDB nicht unterstützt wird.

Um einen Klon eines DB-Clusters zu erstellen, der Ihrem AWS Konto gehört, verwenden Sie AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie Ihren Amazon DocumentDB-Cluster aus der Liste aus und wählen Sie für Aktionen die Option Create clone aus.

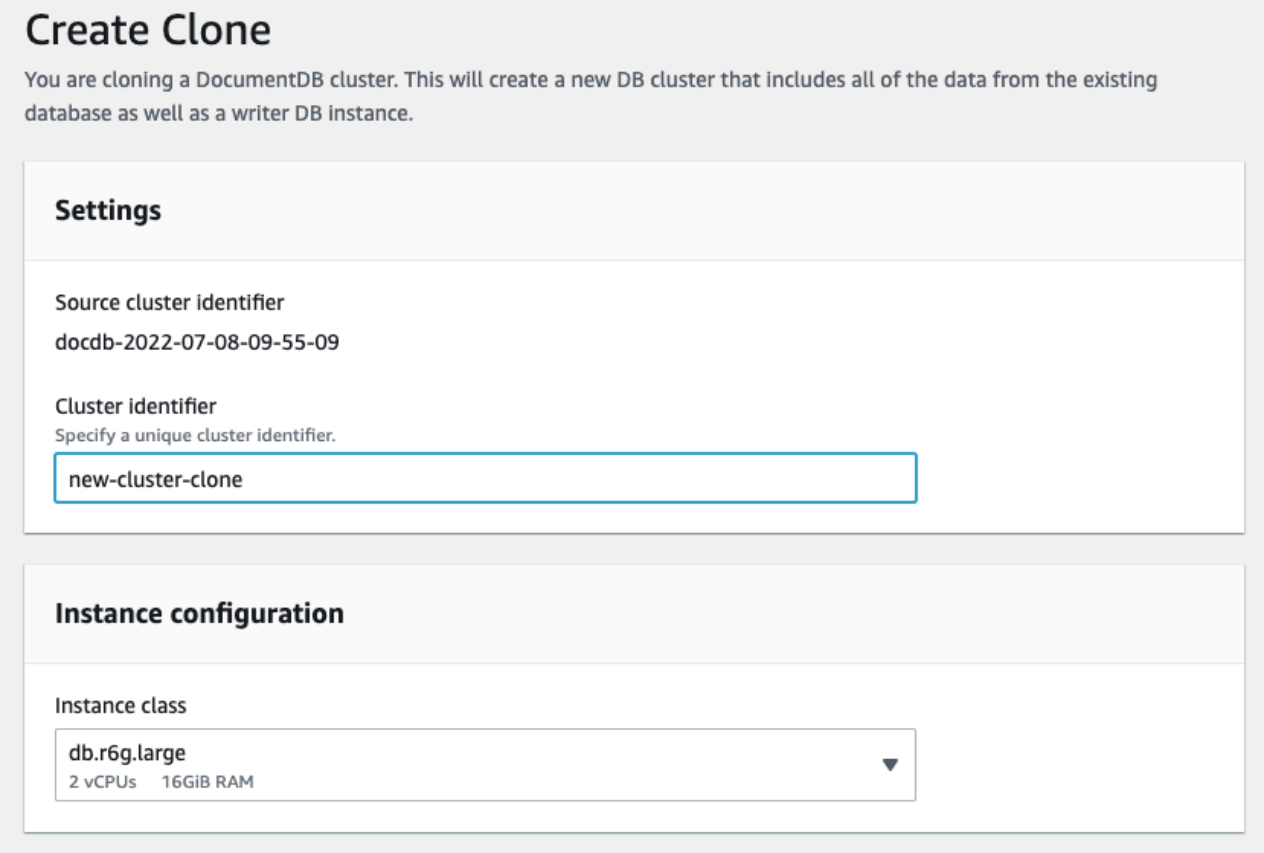


Die Seite „Clone erstellen“ wird geöffnet, auf der Sie eine Cluster-ID und eine Instance-Klasse sowie andere Optionen für den Amazon DocumentDB-Cluster-Klon konfigurieren können.

4. Gehen Sie im Abschnitt Settings (Einstellungen) wie folgt vor:

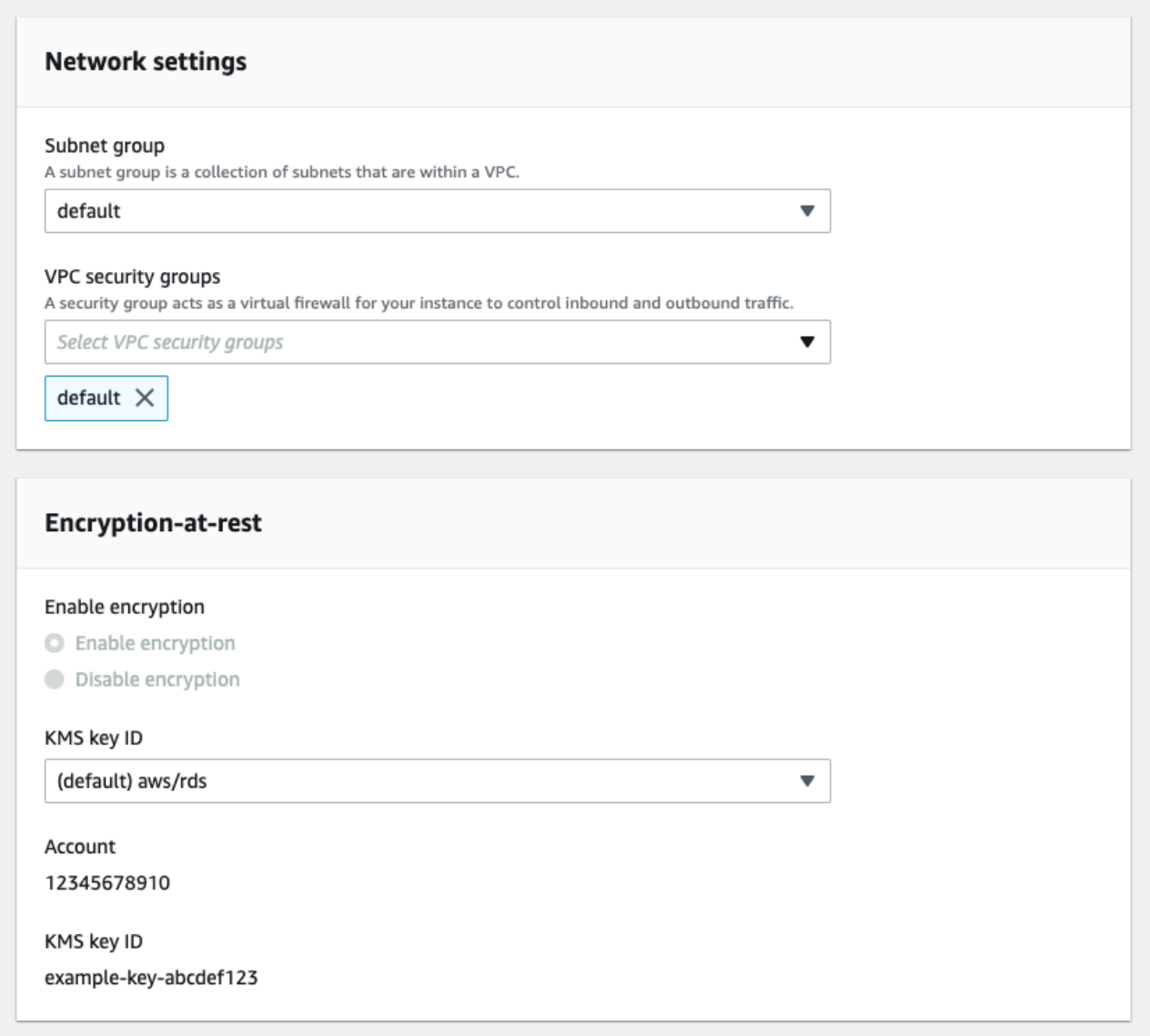
   a. Geben Sie unter Cluster-ID den Namen ein, den Sie Ihrem geklonten Amazon DocumentDB-Cluster geben möchten.

b. Wählen Sie für die Instance-Konfiguration eine geeignete Instance-Klasse für Ihren geklonten Amazon DocumentDB-Cluster aus.

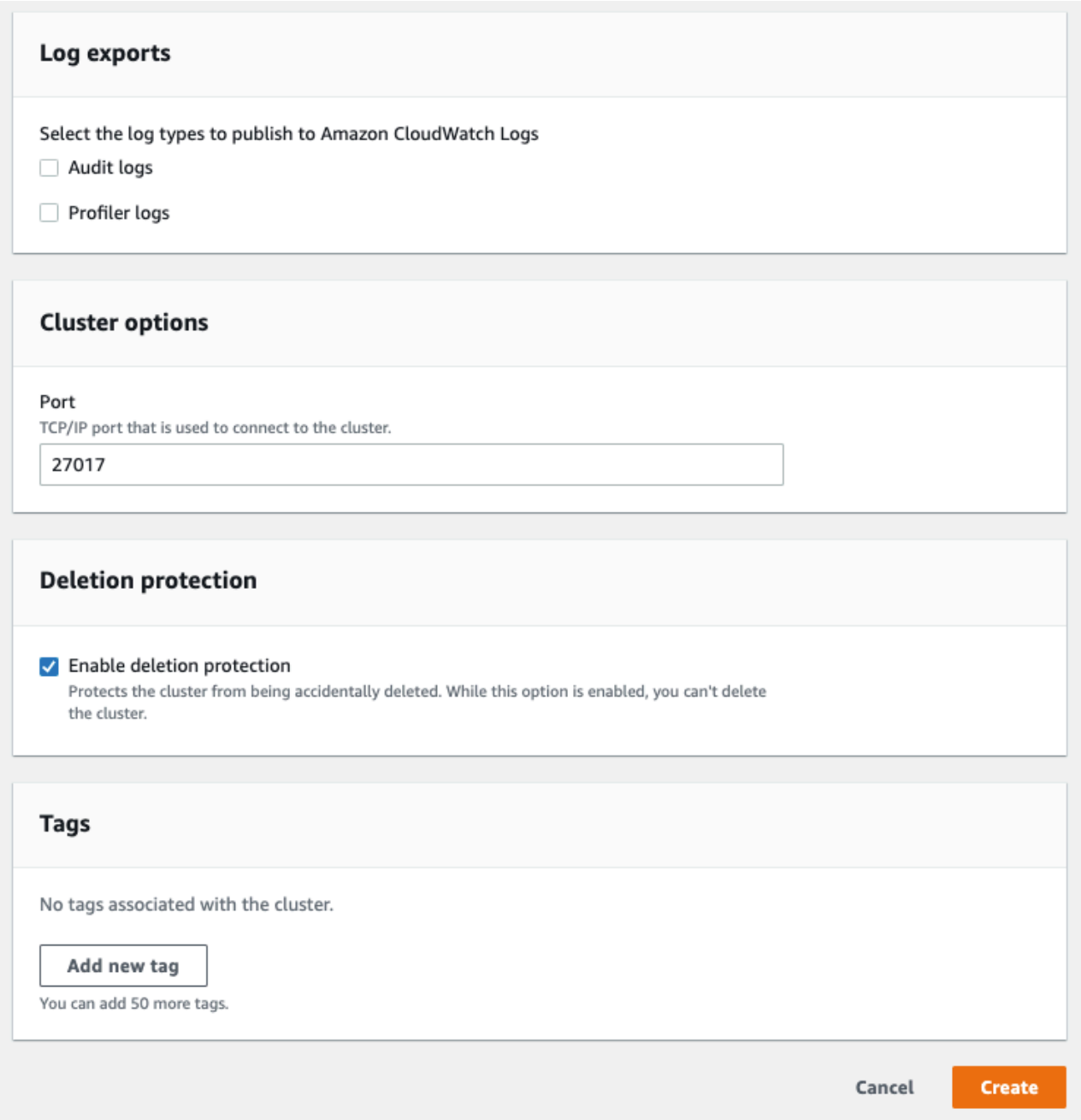


c. Wählen Sie für Netzwerkeinstellungen eine Subnetzgruppe für Ihren Anwendungsfall und die zugehörigen VPC-Sicherheitsgruppen aus.

d. Wenn für E ncryption-at-rest die Verschlüsselung für den Quellcluster (der Cluster, der geklont wird) aktiviert ist, muss auch für den geklonten Cluster die Verschlüsselung aktiviert sein. Wenn dieses Szenario zutrifft, sind die Optionen Verschlüsselung aktivieren ausgegraut (deaktiviert), aber die Option Verschlüsselung aktivieren ist ausgewählt. Umgekehrt, wenn für den Quellcluster keine Verschlüsselung aktiviert ist, sind die Optionen Verschlüsselung aktivieren verfügbar und Sie können wählen, ob die Verschlüsselung aktiviert oder deaktiviert werden soll.



e. Vervollständigen Sie die neue Cluster-Clone-Konfiguration, indem Sie den Typ der zu exportierenden Protokolle auswählen (optional), einen bestimmten Port eingeben, der für die Verbindung mit dem Cluster verwendet wird, und den Schutz vor versehentlichem Löschen des Clusters aktivieren (standardmäßig aktiviert).



f. Beenden Sie die Eingabe aller Einstellungen für Ihren Amazon DocumentDB-Cluster-Klon. Weitere Informationen zu den Cluster- und Instance-Einstellungen von Amazon DocumentDB finden Sie unter[Verwaltung von Amazon DocumentDB-Clustern](#).

5. Wählen Sie Create clone, um den Amazon DocumentDB-Klon Ihres ausgewählten Amazon DocumentDB-Clusters zu starten.

Wenn der Klon erstellt wurde, wird er zusammen mit Ihren anderen Amazon DocumentDB-Clustern im Bereich Datenbanken der Konsole aufgeführt und zeigt seinen aktuellen Status an. Ihr Klon ist einsatzbereit, wenn sein Status Verfügbar ist.

Using the AWS CLI

Die Verwendung des AWS CLI zum Klonen Ihres Amazon DocumentDB-Clusters umfasst einige Schritte.

Der `restore-db-cluster-to-point-in-time` AWS CLI Befehl, den Sie verwenden, führt zu einem leeren Amazon DocumentDB-Cluster mit 0 Amazon DocumentDB DocumentDB-Instances. Das heißt, der Befehl stellt nur den Amazon DocumentDB-Cluster wieder her, nicht die DB-Instances für diesen Cluster. Sie tun dies separat, nachdem der Klon verfügbar ist. Die zwei Schritte im Prozess sind wie folgt:

1. Erstellen Sie den Klon mit dem Befehl restore-db-cluster-to- point-in-time CLI. Die Parameter, die Sie mit diesem Befehl verwenden, steuern den Kapazitätstyp und andere Details des leeren Amazon DocumentDB-Clusters (Klons), der erstellt wird.
2. Erstellen Sie die Amazon DocumentDB-Instance für den Clone, indem Sie den create-db-instanceCLI-Befehl verwenden, um die Amazon DocumentDB DocumentDB-Instance im wiederhergestellten Amazon DocumentDB-Cluster neu zu erstellen.

Bei den folgenden Befehlen AWS CLI wird davon ausgegangen, dass Ihre AWS Region standardmäßig eingerichtet ist. Dieser Ansatz erspart Ihnen die Übergabe des `--region`-Namens in jedem der Befehle. Weitere Informationen finden Sie unter Konfigurieren der AWS CLI. Sie können die `--region` auch in jedem der folgenden CLI-Befehle angeben.

Der Klon wird erstellt

Die spezifischen Parameter, die Sie an den `restore-db-cluster-to-point-in-time`-CLI-Befehl übergeben, variieren. Was Sie übergeben, hängt von der Art des Klons ab, den Sie erstellen möchten.

Gehen Sie wie folgt vor, um einen bereitgestellten Amazon DocumentDB-Klon aus einem bereitgestellten Amazon DocumentDB-Cluster zu erstellen.

Um einen Klon desselben Engine-Modus wie der Amazon DocumentDB DocumentDB-Quellcluster zu erstellen

- Verwenden Sie den `restore-db-cluster-to-point-in-time`-CLI-Befehl und geben Sie Werte für die folgenden Parameter an:

  - `--db-cluster-identifier` – Wählen Sie einen aussagekräftigen Namen für Ihren Klon. Sie geben dem Klon einen Namen, wenn Sie den Befehl restore-db-cluster-to- point-in-time CLI verwenden.
  - `--restore-type` – Verwenden Sie `copy-on-write`, um einen Klon des Quell-DB-Clusters zu erstellen. Ohne diesen Parameter `restore-db-cluster-to-point-in-time` stellt der den Amazon DocumentDB-Cluster wieder her, anstatt einen Klon zu erstellen. Die Standardeinstellung für `restore-type` ist`full-copy`.
  - `--source-db-cluster-identifier`— Verwenden Sie den Namen des Amazon DocumentDB DocumentDB-Quellclusters, den Sie klonen möchten.
  - `--use-latest-restorable-time` – Dieser Wert verweist auf die neuesten wiederherstellbaren Volume-Daten für den Klon. Dieser Parameter ist erforderlich für`restore-type copy-on-write`, Sie können den jedoch nicht `restore-to-time parameter` zusammen verwenden.

Im folgenden Beispiel wird ein Klon namens `my-clone` aus einem Cluster namens `my-source-cluster` erstellt.

Für Linux, macOS oder Unix:

```
aws docdb restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier my-source-cluster \
    --db-cluster-identifier my-clone \
    --restore-type copy-on-write \
    --use-latest-restorable-time
```

Für Windows:

```
aws docdb restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier my-source-cluster ^
    --db-cluster-identifier my-clone ^
    --restore-type copy-on-write ^
    --use-latest-restorable-time
```

Der Befehl gibt das JSON-Objekt zurück, das Details des Klons enthält. Stellen Sie sicher, dass Ihr geklonter DB-Cluster verfügbar ist, bevor Sie versuchen, die DB-Instance für Ihren Klon zu erstellen. Weitere Informationen finden Sie weiter unten unter Status überprüfen und Details zum Klonen abrufen:

Status überprüfen und Details zum Klonen abrufen

Mit dem folgenden Befehl können Sie den Status Ihres neu erstellten leeren DB-Clusters überprüfen.

```
$ aws docdb describe-db-clusters --db-cluster-identifier my-clone --query '*[].
[Status]' --output text
```

Oder Sie können den Status und die anderen Werte, die Sie zum Erstellen der DB-Instance für Ihren Clone benötigen, mithilfe der folgenden AWS CLI Abfrage abrufen:

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone \
  --query '*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}'
```

Für Windows:

```
aws docdb describe-db-clusters --db-cluster-identifier my-clone ^
  --query "*[].{Status:Status,Engine:Engine,EngineVersion:EngineVersion}"
```

Diese Abfrage gibt eine Ausgabe ähnlich der folgenden zurück.

```
[
  {
        "Status": "available",
        "Engine": "docdb",
        "EngineVersion": "4.0.0",
    }
]
```

Erstellen der Amazon DocumentDB DocumentDB-Instance für Ihren Clone

Verwenden Sie den create-db-instanceCLI-Befehl, um die DB-Instance für Ihren Clone zu erstellen.

Der `--db-instance-class` Parameter wird nur für bereitgestellte Amazon DocumentDB-Cluster verwendet.

Für Linux, macOS oder Unix:

```
aws docdb create-db-instance \
    --db-instance-identifier my-new-db \
    --db-cluster-identifier my-clone \
    --db-instance-class  db.r5.4xlarge \
    --engine docdb
```

Für Windows:

```
aws docdb create-db-instance ^
    --db-instance-identifier my-new-db ^
    --db-cluster-identifier my-clone ^
    --db-instance-class  db.r5.4xlarge ^
    --engine docdb
```

Parameter, die für das Klonen verwendet werden sollen

In der folgenden Tabelle sind die verschiedenen Parameter zusammengefasst, die zum Klonen von Amazon DocumentDB-Clustern verwendet werden. `restore-db-cluster-to-point-in-time`

Parameter	Beschreibung
--source-db-cluster-identifier	Verwenden Sie den Namen des Amazon DocumentDB DocumentDB-Quellclusters, den Sie klonen möchten.
--db-cluster-identifier	Wählen Sie einen aussagekräftigen Namen für Ihren Klon. Sie benennen Ihren Klon mit dem `restore-db-cluster-to-point-in-time` -Befehl. Dann übergeben Sie diesen Namen an den `create-db-instance` -Befehl.
--restore-type	Geben Sie `copy-on-write` als an`--restore-type` , um einen Klon des Quell-DB-Clusters zu erstellen, anstatt den Amazon DocumentDB DocumentDB-Quellcluster wiederherzustellen.

Parameter	Beschreibung
--use-latest-restorable-time	ieser Wert verweist auf die neuesten wiederherstellbaren Volume-Daten für den Klon.

Grundlegendes zur Amazon DocumentDB-Cluster-Fehlertoleranz

Amazon DocumentDB-Cluster sind von Natur aus fehlertolerant. Das Volume jedes Clusters erstreckt sich über mehrere Availability Zones in einem einzigen AWS-Region, und jede Availability Zone enthält eine Kopie der Volume-Daten des Clusters. Diese Funktionalität bedeutet, dass Ihr Cluster einen Ausfall der Availability Zone ohne Datenverlust und nur eine kurze Unterbrechung des Services tolerieren kann.

Wenn die primäre Instance in einem Cluster ausfällt, führt Amazon DocumentDB auf zwei Arten automatisch einen Failover zu einer neuen primären Instance durch:

- Indem Sie ein vorhandenes Amazon DocumentDB DocumentDB-Replikat auf die neue primäre Instance heraufstufen, die auf der Grundlage der Promotion-Kontingent-Einstellung für jedes Replikat ausgewählt wurde, und dann einen Ersatz für die frühere primäre Instance erstellen. Ein Failover zur Replikat-Instance dauert in der Regel weniger als 30 Sekunden. Während dieses Zeitraums kann es bei Lese- und Schreibvorgängen zu kurzen Unterbrechungen kommen. Um die Verfügbarkeit Ihres Clusters zu erhöhen, empfehlen wir Ihnen, mindestens ein oder mehrere Amazon DocumentDB DocumentDB-Replikate in zwei oder mehr verschiedenen Availability Zones zu erstellen.
- Über das Anlegen einer neuen primären Instance. Dies passiert nur, wenn Sie keine Replikatinstanz in Ihrem Cluster haben und es kann einige Minuten dauern, bis der Vorgang abgeschlossen ist.

Wenn der Cluster über ein oder mehrere Amazon DocumentDB DocumentDB-Replikate verfügt, wird ein Amazon DocumentDB DocumentDB-Replikat bei einem Ausfall zur primären Instance heraufgestuft. Ein Fehlerereignis hat eine kurze Unterbrechung zufolge, während die Lese- und Schreibvorgänge mit einer Ausnahme fehlschlagen. Jedoch wird der Service im Normalfall in weniger als 120 Sekunden und oft sogar schon nach 60 Sekunden wiederhergestellt. Um die Verfügbarkeit Ihres Clusters zu erhöhen, empfehlen wir Ihnen, mindestens ein oder mehrere Amazon DocumentDB DocumentDB-Replikate in zwei oder mehr verschiedenen Availability Zones zu erstellen.

Sie können die Reihenfolge anpassen, in der Ihre Amazon DocumentDB DocumentDB-Replikate nach einem Ausfall zur primären Instance heraufgestuft werden, indem Sie jedem Replikat eine Priorität zuweisen. Prioritäten liegen im Bereich zwischen 0 als höchste Priorität und 15 als niedrigste Priorität. Wenn die primäre Instance ausfällt, wird das Amazon DocumentDB DocumentDB-Replikat mit der höchsten Priorität zur neuen primären Instance heraufgestuft. Sie können die Priorität eines Amazon DocumentDB DocumentDB-Replikats jederzeit ändern. Das Ändern der Priorität löst kein Failover aus. Sie können die `modify-db-instance`-Operation mit dem `--promotion-tier`-Parameter verwenden. Weitere Informationen zum Anpassen der Failover-Priorität einer Instance finden Sie unter Amazon DocumentDB-Failover.

Mehr als ein Amazon DocumentDB DocumentDB-Replikat kann dieselbe Priorität haben, was zu Aktionsstufen führt. Wenn zwei oder mehr Amazon DocumentDB DocumentDB-Replikate dieselbe Priorität haben, wird das Replikat mit der größten Größe zum primären Replikat heraufgestuft. Wenn zwei oder mehr Amazon DocumentDB DocumentDB-Replikate dieselbe Priorität und Größe haben, wird ein beliebiges Replikat in derselben Promotion-Stufe heraufgestuft.

Wenn der Cluster keine Amazon DocumentDB DocumentDB-Replikate enthält, wird die primäre Instance bei einem Ausfall neu erstellt. Ein Fehlerereignis hat eine Unterbrechung zufolge, während die Lese- und Schreibvorgänge mit einer Ausnahme fehlschlagen. Der Service wird wiederhergestellt, wenn die primäre Instance erstellt wird. Dies dauert im Normalfall weniger als 10 Minuten. Das Heraufstufen eines Amazon DocumentDB-Replikats zur primären Instance ist viel schneller als das Erstellen einer neuen primären Instance.

Verwaltung von Amazon DocumentDB DocumentDB-Instances

Die folgenden Themen enthalten Informationen, die Sie bei der Verwaltung Ihrer Amazon DocumentDB DocumentDB-Instances unterstützen. Sie umfassen Details über Instance-Klassen und Statusarten und das Erstellen, Löschen und Ändern einer Instance.

Themen

- Den Status einer Instance ermitteln
- Lebenszyklus einer Amazon DocumentDB DocumentDB-Instanz
- Instanzklassen verwalten
- NVMe-gestützte Instances

Den Status einer Instance ermitteln

Mehr zu den gültigen Instance-Zuständen, ihre Bedeutung und die Bestimmung des Status Ihrer Instances finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Lebenszyklus einer Amazon DocumentDB DocumentDB-Instanz

Der Lebenszyklus einer Amazon DocumentDB DocumentDB-Instance umfasst das Erstellen, Ändern, Warten und Aktualisieren, das Durchführen von Backups und Wiederherstellungen, das Neustarten und Löschen der Instance. Dieser Abschnitt enthält Informationen darüber, wie Sie diese Prozesse abschließen können.

Themen

- Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster
- Beschreibung von Amazon DocumentDB DocumentDB-Instances
- Ändern einer Amazon DocumentDB DocumentDB-Instance
- Eine Amazon DocumentDB DocumentDB-Instance neu starten
- Löschen einer Amazon DocumentDB DocumentDB-Instance

Sie können eine neue Amazon DocumentDB DocumentDB-Instance mit dem AWS Management Console oder dem AWS CLI erstellen. Um eine Instance zu einem Cluster hinzuzufügen, muss sich der Cluster im Zustand available befinden. Zu einem angehaltenen Cluster können keine Instances hinzugefügt werden. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis der Cluster verfügbar wird, und fügen Sie dann eine Instance hinzu. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster stoppen und starten.

> **Note**
>
> Wenn Sie mit der Konsole einen Amazon DocumentDB-Cluster erstellen, wird gleichzeitig automatisch eine Instance für Sie erstellt. Wenn Sie zusätzliche Instances erstellen möchten, verwenden Sie eines der folgenden Verfahren.

Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster

Using the AWS Management Console

Gehen Sie wie folgt vor, um mithilfe der Amazon DocumentDB DocumentDB-Konsole eine Instance für Ihren Cluster zu erstellen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Um den Cluster zu wählen, dem Sie eine Instance hinzufügen möchten, wählen Sie die Schaltfläche links vom Namen des Clusters aus.
4. Wählen Sie Actions (Aktionen) und dann Add instances (Instances hinzufügen) aus.
5. Wiederholen Sie die folgenden Schritte für jede Instance, die Sie dem Cluster hinzufügen möchten, auf der Seite Add instance to: (Instance hinzufügen zu:)<cluster-name>. Sie können bis zu 15 haben.

   a. Instance-ID — Sie können entweder eine eindeutige Kennung für diese Instance eingeben oder Amazon DocumentDB erlauben, die Instance-ID auf der Grundlage der Cluster-ID bereitzustellen.

      Einschränkungen für Instance-Benennungen:

      - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
      - Muss für alle Instances in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.

b. Instance-Klasse — Wählen Sie aus der Drop-down-Liste den Instance-Typ aus, den Sie für diese Instance benötigen.

c. Promotion-Stufe — Wählen Sie aus der Drop-down-Liste die Promotion-Stufe für Ihre Instance oder wählen Sie Keine Präferenz, damit Amazon DocumentDB die Promotion-Stufe für Ihre Instance festlegen kann. Niedrigere Nummern bedeuten eine höhere Priorität. Weitere Informationen finden Sie unter Steuerung des Failover-Ziels.

d. Zum Hinzufügen weiterer Instances wählen Sie die Option Add additional instances (Weitere Instances hinzufügen) und wiederholen Sie die Schritte a, b und c.

6. Schließen Sie den Vorgang ab.

   - Wenn Sie die Instances zu Ihrem Cluster hinzufügen möchten, klicken Sie auf Create (Erstellen).
   - Um die Operation abzubrechen, wählen Sie Abbrechen aus.

Es dauert einige Minuten, bis eine Instance erstellt ist. Sie können die Konsole verwenden oder AWS CLI den Status der Instance einsehen. Weitere Informationen finden Sie unter Den Status einer Instance überwachen.

Using the AWS CLI

Verwenden Sie den `create-db-instance` AWS CLI Vorgang mit den folgenden Parametern, um die primäre Instance für Ihren Cluster zu erstellen.

- **`--db-instance-class`** – Erforderlich. Die Rechen- und Speicherkapazität der Instance, beispielsweise `db.m4.large`. Nicht alle Instanzklassen sind in allen verfügbar AWS-Regionen.
- **`--db-instance-identifier`** – Erforderlich. Eine -Zeichenfolge in Kleinbuchstaben, die die Instance bezeichnet.

  Instance-Benennungseinschränkungen:

  - Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
  - Muss für alle Instances in Amazon RDS, Neptune und Amazon DocumentDB pro AWS-Konto Region eindeutig sein.
- **`--engine`** – Erforderlich. Der Wert muss `docdb` sein.

- **`--availability-zone`**— Fakultativ. Die Availability Zone, in der diese Instance erstellt werden soll. Verwenden Sie diesen Parameter, um Ihre Instances in verschiedenen Availability Zones zur Erhöhung der Fehlertoleranz zu finden. Weitere Informationen finden Sie unter Amazon DocumentDB Hochverfügbarkeit und Replikation.
- **`--promotion-tier`**— Fakultativ. Die Failover-Prioritätsstufe für diese Instance. Muss zwischen 0 und 15 liegen, wobei niedrigere Nummern eine höhere Priorität bedeuten. Weitere Informationen finden Sie unter Steuerung des Failover-Ziels.

1. Bestimmen Sie zunächst, in welchen Availability Zones Sie Ihre Instance erstellen können.

   Wenn Sie die Availability Zone angeben möchten, bevor Sie Ihre Instance erstellen, führen Sie den folgenden Befehl aus, um zu ermitteln, welche Availability Zones für Ihren Amazon DocumentDB-Cluster verfügbar sind.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-clusters \
         --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
   ```

   Für Windows:

   ```
   aws docdb describe-db-clusters ^
         --query 'DBClusters[*].[DBClusterIdentifier,AvailabilityZones[*]]'
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

   ```
   [
       [
           "sample-cluster",
           [
               "us-east-1c",
               "us-east-1b",
               "us-east-1a"
           ]
       ]
   ]
   ```

2. Bestimmen Sie dann, welche Instance-Klassen Sie in Ihrer Region erstellen können.

Führen Sie den folgenden Befehl aus, um festzustellen, welche Instance-Klassen Ihnen in Ihrer Region zur Verfügung stehen. Wählen Sie aus der Ausgabe eine Instance-Klasse für die Instance aus, die Sie Ihrem Amazon DocumentDB-Cluster hinzufügen möchten.

Für Linux, macOS oder Unix:

```
aws docdb describe-orderable-db-instance-options \
        --engine docdb \
        --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

Für Windows:

```
aws docdb describe-orderable-db-instance-options ^
        --engine docdb ^
        --query 'OrderableDBInstanceOptions[*].DBInstanceClass'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    "db.r5.16xlarge",
    "db.r5.2xlarge",
    "db.r5.4xlarge",
    "db.r5.8xlarge",
    "db.r5.large",
    "db.r5.xlarge"
]
```

3. Fügen Sie abschließend eine Instance zu Ihrem Amazon DocumentDB-Cluster hinzu.

   Um Ihrem Amazon DocumentDB-Cluster eine Instance hinzuzufügen, führen Sie den folgenden Befehl aus.

   Für Linux, macOS oder Unix:

```
aws docdb create-db-instance \
       --db-cluster-identifier sample-cluster \
       --db-instance-identifier sample-instance-2 \
       --availability-zone us-east-1b \
       --promotion-tier 2 \
       --db-instance-class db.r5.xlarge \
```

```
        --engine docdb
```

Für Windows:

```
aws docdb create-db-instance ^
       --db-cluster-identifier sample-cluster ^
       --db-instance-identifier sample-instance-2 ^
       --availability-zone us-east-1b ^
       --promotion-tier 2 ^
       --db-instance-class db.r5.xlarge ^
       --engine docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance-2",
        "DBInstanceClass": "db.r5.xlarge",
        "Engine": "docdb",
        "DBInstanceStatus": "creating",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1b",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
```

```
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:11:35-sun:12:05",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 2,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2"
    }
}
```

Es dauert einige Minuten, bis die Instance erstellt ist. Sie können die Konsole verwenden oder AWS CLI den Status der Instance einsehen. Weitere Informationen finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Beschreibung von Amazon DocumentDB DocumentDB-Instances

Sie können entweder die Amazon DocumentDB-Managementkonsole oder die verwenden, AWS CLI um Details wie Verbindungsendpunkte, Sicherheitsgruppen VPCs, Zertifizierungsstellen und Parametergruppen zu Ihren Amazon DocumentDB DocumentDB-Instances anzuzeigen.

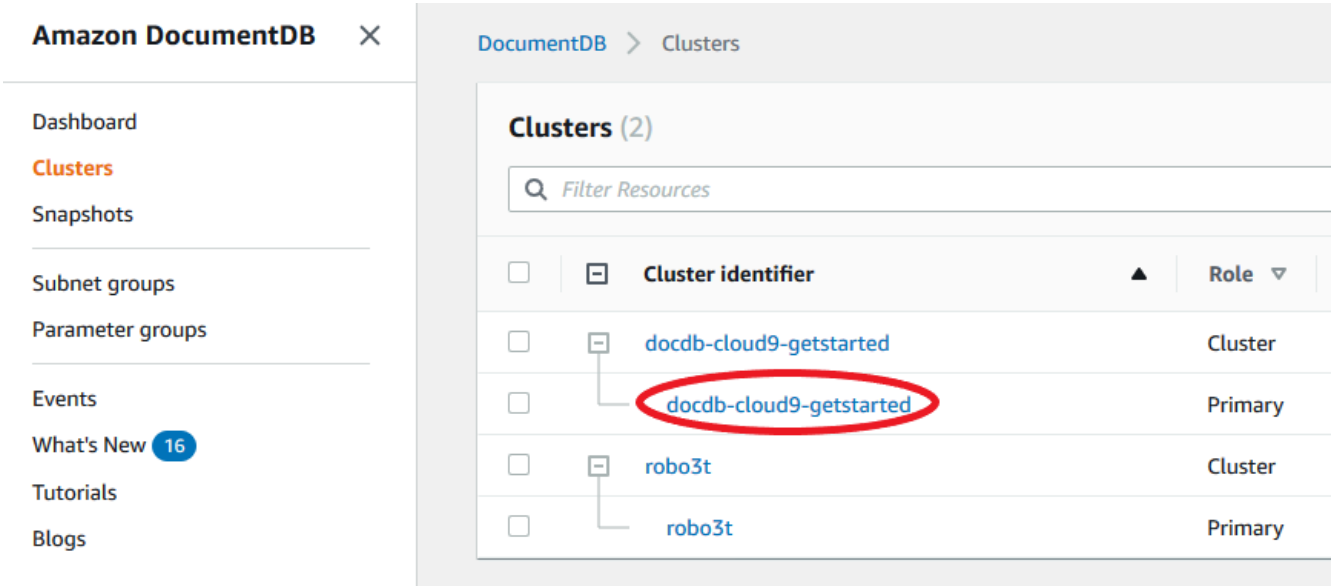
Using the AWS Management Console

Gehen Sie folgendermaßen vor, um die Details Ihrer Instances mithilfe der AWS Management Console anzuzeigen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Wählen Sie in der Liste der Instances den Namen der Instanz aus, deren Details Sie sehen möchten. Die Informationen über die Instance sind in die folgenden Gruppierungen unterteilt:

   - Zusammenfassung — Allgemeine Informationen über die Instanz, einschließlich der Engine-Version, der Klasse, des Status und aller ausstehenden Wartungsarbeiten.
   - Konnektivität und Sicherheit — Der Abschnitt Connect listet die Verbindungsendpunkte auf, um mit der Mongo-Shell oder mit einer Anwendung eine Verbindung zu dieser Instanz herzustellen. Im Abschnitt Sicherheitsgruppen werden die Sicherheitsgruppen aufgeführt, die dieser Instance zugeordnet sind, sowie ihre VPC-ID und Beschreibungen.
   - Konfiguration — Der Abschnitt „Details“ listet die Konfigurationen und den Status der Instance auf, einschließlich des Amazon-Ressourcennamens (ARN), des Endpunkts,

der Rolle, der Klasse und der Zertifizierungsstelle der Instance. Außerdem werden die Sicherheits- und Netzwerkeinstellungen der Instance sowie die Sicherungsinformationen aufgelistet. Im Abschnitt Cluster-Details werden die Details des Clusters aufgeführt, zu dem diese Instance gehört. Im Abschnitt Cluster-Instances werden alle Instances aufgeführt, die zu Ihrem Cluster gehören, mit dem Status der einzelnen Instance-Rollen und Cluster-Parametergruppen.

> **Note**
>
> Sie können den mit Ihrer Instance verknüpften Cluster ändern, indem Sie neben der Überschrift Cluster-Details auf Ändern klicken. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

- Überwachung — Die CloudWatch Log-Metriken für diese Instance. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.
- Ereignisse und Tags — Der Abschnitt Aktuelle Ereignisse listet die jüngsten Ereignisse für diese Instance auf. Amazon DocumentDB zeichnet Ereignisse auf, die sich auf Ihre Cluster, Instances, Snapshots, Sicherheitsgruppen und Cluster-Parametergruppen beziehen. Zu diesen Informationen gehören Datum, Uhrzeit und Nachricht, die jedem Ereignis zugeordnet sind. Der Abschnitt Tags listet die an diesen Cluster angehängten Tags auf. Weitere Informationen finden Sie unter Taggen von Amazon DocumentDB DocumentDB-Ressourcen.

Using the AWS CLI

Um die Details Ihrer Amazon DocumentDB DocumentDB-Instances mithilfe von anzuzeigen AWS CLI, verwenden Sie den `describe-db-clusters` Befehl, wie in den folgenden Beispielen gezeigt. Weitere Informationen finden Sie `DescribeDBInstances`in der Amazon DocumentDB Resource Management API-Referenz.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen wie Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird. Der `filterName=engine,Values=docdb` Filterparameter gibt nur Amazon DocumentDB-Cluster zurück.

1. Listet alle Amazon DocumentDB DocumentDB-Instances auf.

   Der folgende AWS CLI Code listet die Details für alle Amazon DocumentDB DocumentDB-Instances in einer Region auf.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-instances \
       --filter Name=engine,Values=docdb
   ```

   Für Windows:

   ```
   aws docdb describe-db-instances \
       --filter Name=engine,Values=docdb
   ```

2. Alle Details für eine angegebene Amazon DocumentDB DocumentDB-Instance auflisten

   Der folgende Code listet die Details für `sample-cluster-instance` auf. Das Einschließen des Parameters `--db-instance-identifier` mit dem Namen einer Instance beschränkt die Ausgabe auf Informationen zu dieser bestimmten Instance.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-instances \
       --db-instance-identifier sample-cluster-instance
   ```

   Für Windows:

   ```
   aws docdb describe-db-instances \
       --db-instance-identifier sample-cluster-instance
   ```

   Die Ausgabe dieser Operation sieht folgendermaßen aus.

   ```
   {
       "DBInstances": [
           {
               "DbiResourceId": "db-BJKKB54PIDV5QFKGVRX5T3S6GM",
               "DBInstanceArn": "arn:aws:rds:us-east-1:012345678901:db:sample-
   cluster-instance-00",
               "VpcSecurityGroups": [
   ```

```
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "DBInstanceClass": "db.r5.large",
        "DBInstanceStatus": "creating",
        "AutoMinorVersionUpgrade": true,
        "PreferredMaintenanceWindow": "fri:09:32-fri:10:02",
        "BackupRetentionPeriod": 1,
        "StorageEncrypted": true,
        "DBClusterIdentifier": "sample-cluster",
        "EngineVersion": "3.6.0",
        "AvailabilityZone": "us-east-1a",
        "Engine": "docdb",
        "PromotionTier": 2,
        "DBInstanceIdentifier": "sample-cluster-instance",
        "PreferredBackupWindow": "00:00-00:30",
        "PubliclyAccessible": false,
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-4e26d263",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-afc329f4",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-b3806e8f",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1e"
                    },
                    "SubnetStatus": "Active"
                },
                {
```

```
                        "SubnetIdentifier": "subnet-53ab3636",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1d"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-991cb8d0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1b"
                        },
                        "SubnetStatus": "Active"
                    },
                    {
                        "SubnetIdentifier": "subnet-29ab1025",
                        "SubnetAvailabilityZone": {
                            "Name": "us-east-1f"
                        },
                        "SubnetStatus": "Active"
                    }
                ],
                "VpcId": "vpc-91280df6",
                "DBSubnetGroupDescription": "default",
                "SubnetGroupStatus": "Complete"
            },
            "PendingModifiedValues": {},
            "KmsKeyId": "arn:aws:kms:us-east-1:012345678901:key/0961325d-
a50b-44d4-b6a0-a177d5ff730b"
        }
    ]
}
```


Ändern einer Amazon DocumentDB DocumentDB-Instance

Sie können Ihre Amazon DocumentDB DocumentDB-Instance entweder mit dem AWS Management Console oder dem AWS CLIändern. Nur Instances im Status available können geändert werden. Eine angehaltene Instance kann nicht geändert werden. Wenn der Cluster angehalten ist, starten Sie zuerst den Cluster, warten Sie, bis die Instance verfügbar wird, und nehmen Sie dann die gewünschten Änderungen vor. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster stoppen und starten.

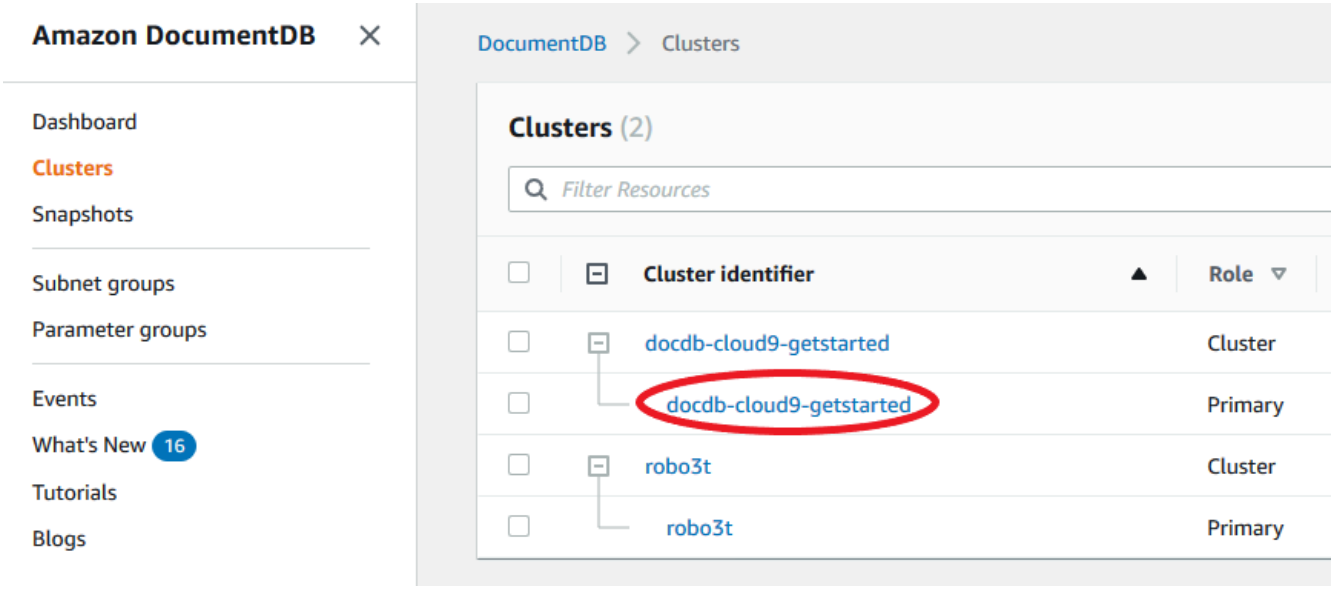
Using the AWS Management Console

Gehen Sie wie folgt vor, um eine bestimmte Amazon DocumentDB DocumentDB-Instance mithilfe der Konsole zu ändern.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Markieren Sie das Kästchen links neben der Instanz, die Sie ändern möchten.
5. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
6. Nehmen Sie im Bereich Modify instance: <instance-name> (Instance ändern: <instance-name>) die gewünschten Änderungen vor. Sie können die folgenden Änderungen ausführen:

   - Instanzspezifikationen — Die Instanz-ID und Klasse. Benennungseinschränkungen für Instance-IDs:

- Instanz-ID — Geben Sie einen Namen ein, der für alle Instances, die Ihnen gehören, AWS-Konto in der aktuellen Region eindeutig ist. Die Instance-ID muss [1—63] alphanumerische Zeichen oder Bindestriche enthalten, einen Buchstaben als erstes Zeichen haben und darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- Instance-Klasse — Wählen Sie im Drop-down-Menü eine Instance-Klasse für Ihre Amazon DocumentDB DocumentDB-Instance aus. Weitere Informationen finden Sie unter Instanzklassen verwalten.

- Zertifizierungsstelle — Serverzertifikat für diese Instance. Weitere Informationen finden Sie unter Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate.
- Failover — Während des Failovers wird die Instanz mit der höchsten Heraufstufungsstufe zur primären Instanz heraufgestuft. Weitere Informationen finden Sie unter Amazon DocumentDB-Failover.
- Wartung — Das Wartungsfenster, in dem ausstehende Änderungen oder Patches auf Instanzen im Cluster angewendet werden.

7. Wenn Sie fertig sind, wählen Sie Continue (Weiter) aus, um eine Zusammenfassung Ihrer Änderungen anzuzeigen.
8. Nachdem Sie Ihre Änderungen überprüft haben, können Sie diese sofort oder während des nächsten Wartungsfensters unter Scheduling of modifications (Planen von Änderungen) anwenden. Klicken Sie auf Modify instance (Instance ändern), um Ihre Änderungen zu speichern. Alternativ können Sie auf Cancel (Abbrechen) klicken, um Ihre Änderungen zu verwerfen.

Es dauert einige Minuten, bis Ihre Änderungen übernommen werden. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Using the AWS CLI

Um eine bestimmte Amazon DocumentDB DocumentDB-Instance mithilfe von zu ändern AWS CLI, verwenden Sie die `modify-db-instance` mit den folgenden Parametern. Weitere Informationen finden Sie unter Ändern DBInstance. Der folgende Code ändert die Instance-Klasse für die Instance `sample-instance` in `db.r5.large`.

Parameter

- **`--db-instance-identifier`** – Erforderlich. Die Kennung der zu ändernden Instance.
- **`--db-instance-class`**— Fakultativ. Die neue Rechen- und Speicherkapazität der Instanz; zum Beispiel`db.r5.large`. Nicht alle Instanzklassen sind in allen verfügbar AWS-Regionen. Wenn Sie die Instanzklasse ändern, kommt es während der Änderung zu einem Ausfall. Die Änderung wird im nächsten Wartungsfenster übernommen, sofern sie `ApplyImmediately` für diese Anfrage nicht als wahr angegeben wurde.
- **`--apply-immediately`**oder **`--no-apply-immediately`** — Optional. Gibt an, ob diese Änderung sofort angewandt oder bis zum nächsten Wartungszeitraum verschoben werden sollte. Wenn dieser Parameter weggelassen wird, wird die Änderung im nächsten Wartungszeitraum ausgeführt.

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-instance \
       --db-instance-identifier sample-instance \
       --db-instance-class db.r5.large \
       --apply-immediately
```

Für Windows:

```
aws docdb modify-db-instance ^
       --db-instance-identifier sample-instance ^
       --db-instance-class db.r5.large ^
       --apply-immediately
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "sample-instance-1",
            "DBInstanceClass": "db.r5.large",
            "Engine": "docdb",
            "DBInstanceStatus": "modifying",
            "Endpoint": {
                "Address": "sample-instance-1.node.us-east-1.docdb.amazonaws.com",
```

```
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-01-10T22:18:55.921Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1a",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-abcd0123",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:10:57-sun:11:27",
        "PendingModifiedValues": {
            "DBInstanceClass": "db.r5.large"
        },
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
```

```
            "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/wJalrXUtnFEMI/
K7MDENG/bPxRfiCYEXAMPLEKEY",
            "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
            "CACertificateIdentifier": "rds-ca-2019",
            "PromotionTier": 1,
            "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:sample-
instance-1",
            "EnabledCloudwatchLogsExports": [
                "profiler"
            ]
        }
    ]
}
```

Es dauert einige Minuten, bis Ihre Änderungen angewendet werden. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können den Status der Instanz mit dem AWS Management Console oder überwachen AWS CLI. Weitere Informationen finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Eine Amazon DocumentDB DocumentDB-Instance neu starten

Gelegentlich müssen Sie Ihre Amazon DocumentDB DocumentDB-Instance möglicherweise neu starten, normalerweise aus Wartungsgründen. Wenn Sie bestimmte Änderungen vornehmen, z. B. die Clusterparametergruppe, die einem Cluster zugeordnet ist, müssen Sie die Instances im Cluster neu starten, damit die Änderungen wirksam werden. Sie können eine angegebene Instance mit dem AWS Management Console oder dem AWS CLI neu starten.

Durch den Neustart einer Instance wird der Datenbank-Engine-Service neu gestartet. Das Neustarten bewirkt einen vorübergehenden Nutzungsausfall, wobei der Status für die Instance währenddessen auf `rebooting` gesetzt wird. Ein Amazon DocumentDB DocumentDB-Ereignis wird erstellt, wenn der Neustart abgeschlossen ist.

Ein Neustart einer Instance führt nicht zu einem Failover. Verwenden Sie die Operation AWS Management Console oder, um ein Failover für einen Amazon DocumentDB-Cluster durchzuführen. AWS CLI `failover-db-cluster` Weitere Informationen finden Sie unter Amazon DocumentDB-Failover.

Sie können Ihre Instance nicht neu starten, wenn sie sich nicht im verfügbaren Zustand befindet. Ihre Datenbank kann aus mehreren Gründen nicht verfügbar sein, zum Beispiel aufgrund einer zuvor

angeforderten Änderung oder einer Aktion im Wartungsfenster. Weitere Informationen zu Instance-Status finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

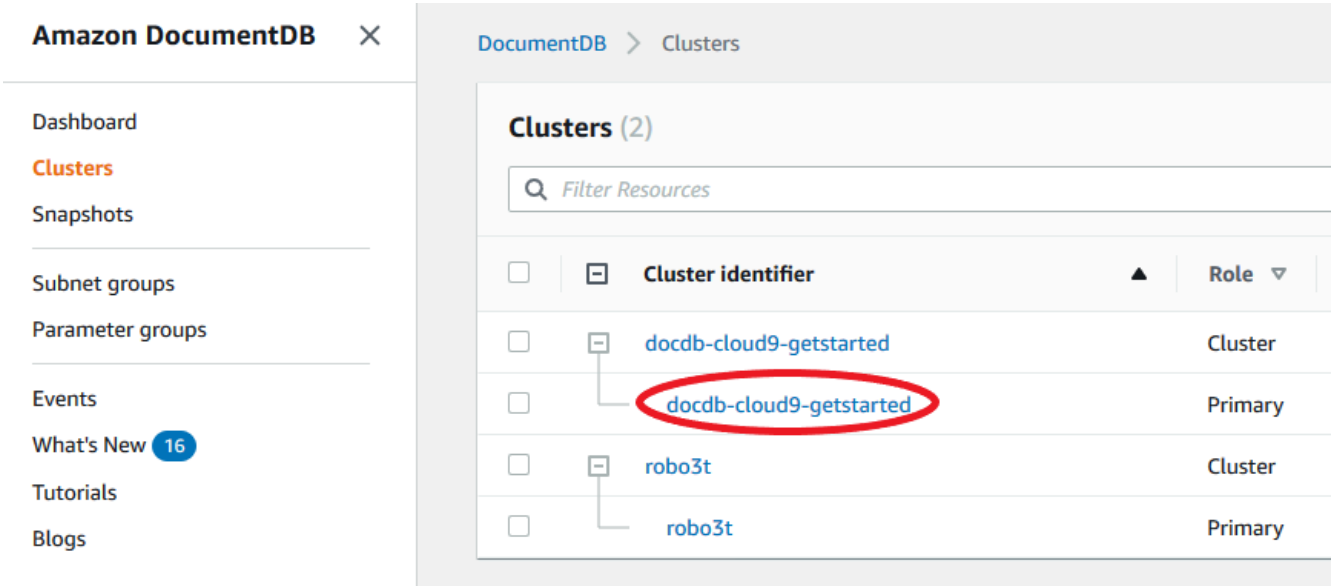
Using the AWS Management Console

Das folgende Verfahren startet eine Instance neu, die Sie unter Verwendung der Konsole angeben.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Markieren Sie das Kästchen links neben der Instance, die Sie neu starten möchten.
5. Wählen Sie Actions (Aktionen), Reboot (Neustart) und dann Reboot (Neustart)aus, um den Neustart zu bestätigen.

Es dauert einige Minuten, bis Ihre Instance neu gestartet wird. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Using the AWS CLI

Um eine Amazon DocumentDB DocumentDB-Instance neu zu starten, verwenden Sie den `reboot-db-instance` Vorgang mit dem `--db-instance-identifier` Parameter. Dieser Parameter gibt den Bezeichner für die Instance an, die neu gestartet werden soll.

Der folgende Code startet die Instance `sample-instance` neu.

Example

Für Linux, macOS oder Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-instance
```

Für Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-instance
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "rebooting",
        "Endpoint": {
            "Address": "sample-instance.node.us-east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
```

```
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1c",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-abcd0123",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ]
    }
}
```

Es dauert einige Minuten, bis Ihre Instance neu gestartet wird. Sie können die Instance nur verwenden, wenn ihr Status available ist. Sie können mit der Konsole oder der AWS CLI den Status der Instance überwachen. Weitere Informationen finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Löschen einer Amazon DocumentDB DocumentDB-Instance

Sie können Ihre Amazon DocumentDB DocumentDB-Instance entweder mit AWS Management Console oder mit dem löschen. AWS CLI Nur Instances im Status available können gelöscht werden. Eine angehaltene Instance kann nicht gelöscht werden. Wenn der Amazon DocumentDB-Cluster, der Ihre Instance enthält, gestoppt ist, starten Sie zuerst den Cluster, warten Sie, bis die Instance verfügbar ist, und löschen Sie dann die Instance. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster stoppen und starten.

> **Note**
>
> Amazon DocumentDB speichert all Ihre Daten im Cluster-Volume. Die Daten bleiben in diesem Cluster-Volume erhalten. Dies gilt auch dann, wenn Sie alle Instances von Ihrem Cluster entfernen. Wenn Sie erneut auf die Daten zugreifen müssen, können Sie jederzeit dem Cluster eine Instance hinzufügen und da fortfahren, wo Sie aufgehört haben.

Using the AWS Management Console

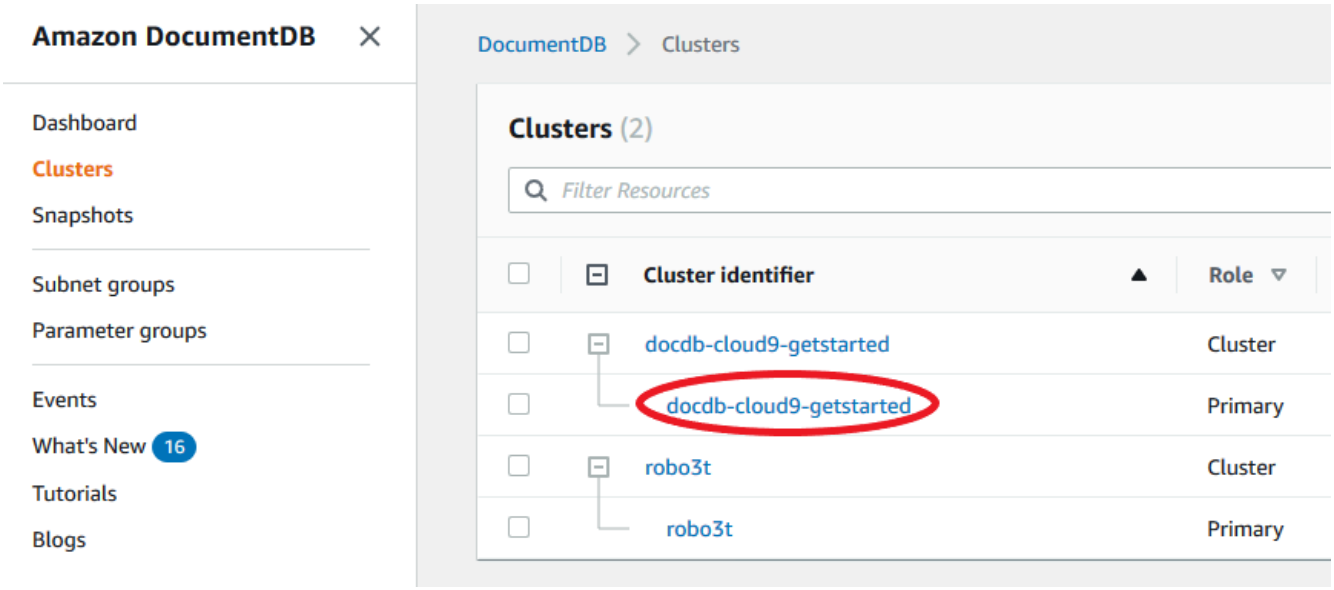
Das folgende Verfahren löscht eine angegebene Amazon DocumentDB DocumentDB-Instance mithilfe der Konsole.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol

(≡)
aus.

3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Markieren Sie das Kästchen links neben der Instance, die Sie löschen möchten.
5. Wählen Sie Actions (Aktionen) und anschließend Delete (Löschen) aus.

   1. Wenn Sie die letzte Instance in Ihrem Cluster löschen:

      - Create final cluster snapshot? (Endgültigen Cluster-Snapshot erstellen?) — Wählen Sie Ja, wenn Sie einen endgültigen Snapshot erstellen möchten, bevor der Cluster gelöscht wird. Wählen Sie andernfalls No (Nein) aus.
      - Name des endgültigen Snapshots — Wenn Sie einen endgültigen Snapshot erstellen möchten, geben Sie die Cluster-Snapshot-ID des neu erstellten Cluster-Snapshots ein.
      - Delete <instance-name> instance? (Instance <instance-name> löschen?) — Geben Sie den Ausdruck Gesamten Cluster löschen in das Feld ein, um das Löschen zu bestätigen.

   2. Wenn Sie nicht die letzte Instance in Ihrem Cluster löschen:

      - Delete <instance-name> instance? (Instance <instance-name> löschen?) — Geben Sie den Ausdruck delete me in das Feld ein, um das Löschen zu bestätigen.

6. Wählen Sie Delete (Löschen), um die Instance zu löschen.

Es dauert einige Minuten, bis die Instance gelöscht ist. Informationen zum Überwachen des Status einer Instance finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

Using the AWS CLI

Das folgende Verfahren löscht eine Amazon DocumentDB DocumentDB-Instance mithilfe von. AWS CLI

1. Stellen Sie zunächst fest, wie viele Instances sich in Ihrem Amazon DocumentDB-Cluster befinden:

   Führen Sie den `describe-db-clusters`-Befehl wie folgt aus, um festzustellen, wie viele Instances sich in Ihrem Cluster befinden.

   ```
   aws docdb describe-db-clusters \
       --db-cluster-identifier sample-cluster \
       --query 'DBClusters[*].
   [DBClusterIdentifier,DBClusterMembers[*].DBInstanceIdentifier]'
   ```

   Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

   ```
   [
       [
           "sample-cluster",
           [
               "sample-instance-1",
               "sample-instance-2"
           ]
       ]
   ]
   ```

2. Wenn es mehr als eine Instance in Ihrem Amazon DocumentDB-Cluster gibt:

   Um eine angegebene Amazon DocumentDB DocumentDB-Instance zu löschen, verwenden Sie den `delete-db-instance` Befehl mit dem `--db-instance-identifier` Parameter, wie unten gezeigt. Es dauert einige Minuten, bis die Instance gelöscht ist. Informationen zum Überwachen des Status einer Instance finden Sie unter Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen.

   ```
   aws docdb delete-db-instance \
   ```

```
        --db-instance-identifier sample-instance-2
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "sample-instance-2",
        "DBInstanceClass": "db.r5.large",
        "Engine": "docdb",
        "DBInstanceStatus": "deleting",
        "Endpoint": {
            "Address": "sample-instance-2.node.us-east-1.docdb.amazonaws.com",
            "Port": 27017,
            "HostedZoneId": "ABCDEFGHIJKLM"
        },
        "InstanceCreateTime": "2020-03-27T08:05:56.314Z",
        "PreferredBackupWindow": "02:00-02:30",
        "BackupRetentionPeriod": 1,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "AvailabilityZone": "us-east-1c",
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-6242c31a",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-abcd0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-wxyz0123",
                    "SubnetAvailabilityZone": {
                        "Name": "us-east-1b"
                    },
```

```
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:06:53-sun:07:23",
        "PendingModifiedValues": {},
        "EngineVersion": "3.6.0",
        "AutoMinorVersionUpgrade": true,
        "PubliclyAccessible": false,
        "DBClusterIdentifier": "sample-cluster",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "CACertificateIdentifier": "rds-ca-2019",
        "PromotionTier": 1,
        "DBInstanceArn": "arn:aws:rds:us-east-1:<accountID>:db:sample-instance-2",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ]
    }
}
```

3. Wenn die Instance, die Sie löschen möchten, die letzte Instance in Ihrem Amazon DocumentDB-Cluster ist:

   Wenn Sie die letzte Instance in einem Amazon DocumentDB-Cluster löschen, löschen Sie auch diesen Cluster und die mit diesem Cluster verknüpften automatischen Snapshots und kontinuierlichen Backups.

   Um die letzte Instance in Ihrem Cluster zu löschen, können Sie den Cluster löschen und optional einen endgültigen Snapshot erstellen. Weitere Informationen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Löschschutz

Wenn Sie die letzte Instanz eines Amazon DocumentDB-Clusters löschen, werden auch der Cluster sowie die mit diesem Cluster verknüpften automatischen Snapshots und kontinuierlichen Backups gelöscht. Amazon DocumentDB erzwingt den Löschschutz für einen Cluster, unabhängig davon, ob Sie den Löschvorgang mit AWS Management Console oder dem ausführen. AWS CLI Sie können einen Cluster nicht löschen, solange der Löschschutz aktiviert ist.

Um einen Cluster mit aktiviertem Löschschutz zu löschen, müssen Sie zuerst den Cluster ändern und den Löschschutz deaktivieren. Weitere Informationen finden Sie unter Löschen eines Amazon DocumentDB-Clusters.

Instanzklassen verwalten

Die Instance-Klasse bestimmt die Rechenleistung und Speicherkapazität einer Amazon DocumentDB DocumentDB-Instance (mit MongoDB-Kompatibilität). Welche Instance-Klasse Sie benötigen, hängt von der benötigten Rechenleistung und dem Speicherbedarf ab.

Amazon DocumentDB unterstützt die Instance-Klassen R4, R5, R6G, T3 und T4G. Diese Klassen sind Instance-Klassen der aktuellen Generation, die für speicherintensive Anwendungen optimiert sind. Die Spezifikationen dieser Klassen finden Sie unter Spezifikationen der Instanzklasse.

Themen

- Bestimmung einer Instance-Klasse
- Die Klasse einer Instance ändern
- Unterstützte Instance-Klassen nach Region
- Spezifikationen der Instanzklasse

Bestimmung einer Instance-Klasse

Um die Klasse einer Instanz zu ermitteln, können Sie die `describe-db-instances` AWS CLI Operation AWS Management Console oder verwenden.

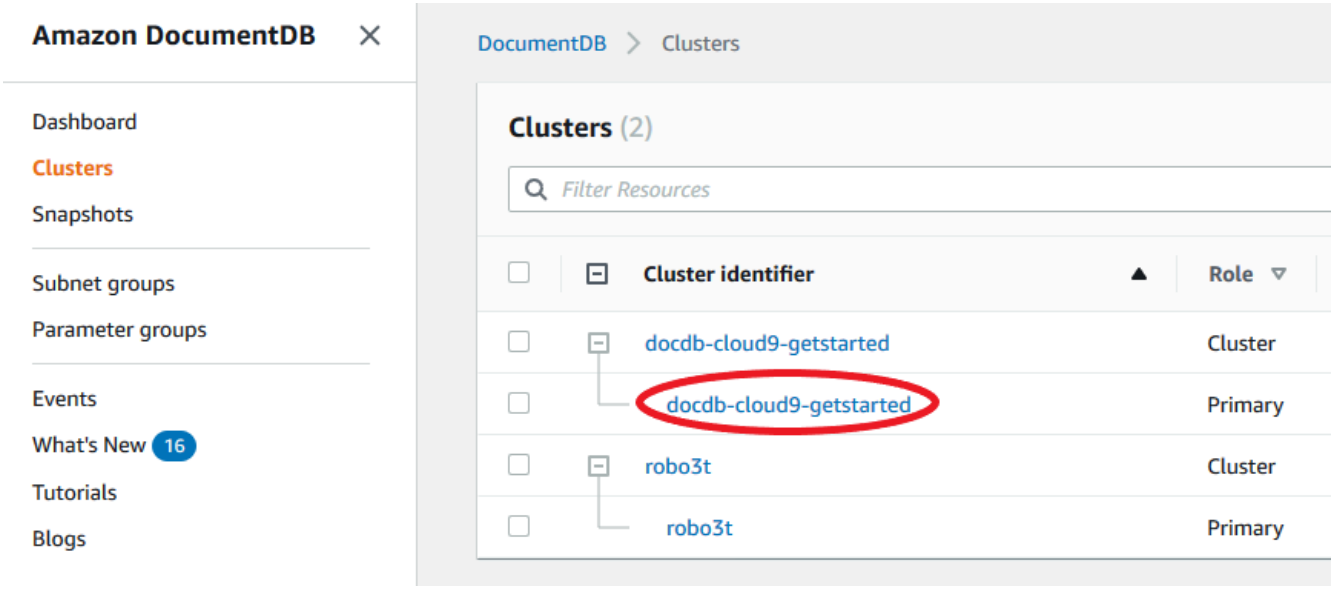
Using the AWS Management Console

Führen Sie die folgenden Schritte in der Konsole aus, um die Instanzklasse für die Instances Ihres Clusters zu ermitteln.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Clusters aus, um die Instance zu finden, an der Sie interessiert sind.

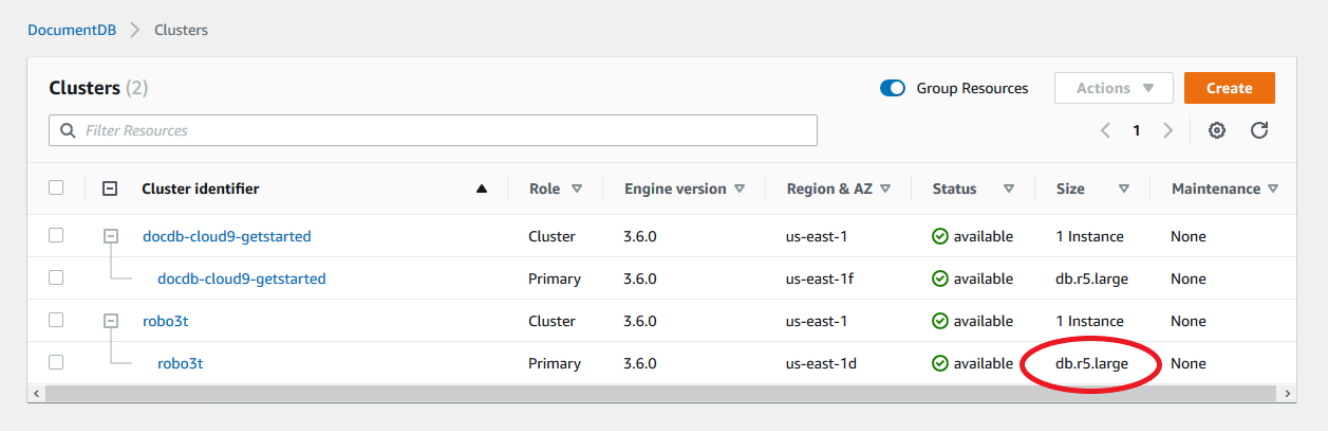
> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Navigationsfeld „Cluster“ sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Erweitern Sie in der Liste der Instances den Cluster, um die Instances zu finden, an denen Sie interessiert sind. Suchen Sie die gewünschte Instanz. Sehen Sie sich dann die Spalte Größe in der Zeile der Instanz an, um deren Instanzklasse zu sehen.

   In der folgenden Abbildung ist die Instance-Klasse für die Instance `robo3t` `db.r5.4xlarge`.



Using the AWS CLI

Verwenden Sie die `describe-db-instances` Operation mit den folgenden Parametern AWS CLI, um die Klasse einer Instance mithilfe von zu ermitteln.

- **`--db-instance-identifier`**— Fakultativ. Gibt die Instance an, für die Sie die Instance-Klasse suchen möchten. Wenn dieser Parameter ausgelassen wird, gibt `describe-db-instances` eine Beschreibung für bis zu 100 Ihrer Instances zurück.
- **`--query`**— Fakultativ. Gibt die Mitglieder der Instance an, die in die Ergebnisse einbezogen werden sollen. Wenn dieser Parameter ausgelassen wird, werden alle Instance-Mitglieder zurückgegeben.

Example

Im folgenden Beispiel werden der Instanzname und die Klasse für die Instanz ermitteltsample-instance-1.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --db-instance-identifier sample-instance-1
```

Für Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --db-instance-identifier sample-instance-1
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ]
```

Example

Das folgende Beispiel ermittelt den Instanznamen und die Klasse für bis zu 100 Amazon DocumentDB DocumentDB-Instances.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' \
    --filter Name=engine,Values=docdb
```

Für Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceClass]' ^
    --filter Name=engine,Values=docdb
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-instance-1",
        "db.r5.large"
    ],
    [
        "sample-instance-2",
        "db.r5.large"
    ],
    [
        "sample-instance-3",
        "db.r5.4xlarge"
    ],
    [
        "sample-instance-4",
```

```
        "db.r5.4xlarge"
    ]
]
```

Weitere Informationen finden Sie unter Beschreibung von Amazon DocumentDB DocumentDB-Instances.

Die Klasse einer Instance ändern

Sie können die Instanzklasse Ihrer Instanz mit dem AWS Management Console oder dem ändern AWS CLI. Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB DocumentDB-Instance.

Unterstützte Instance-Klassen nach Region

Amazon DocumentDB unterstützt die folgenden Instance-Klassen:

- R6G—Speicheroptimierte Instances der neuesten Generation mit ARM-basierten AWS Graviton2-Prozessoren, die im Vergleich zu R5-Instances eine um bis zu 30% bessere Leistung bei 5% geringeren Kosten bieten.
- R6GD—Speicheroptimierte R6G-Instances mit lokalem Solid-State-Drive-Speicher (SSD) auf Basis von nichtflüchtigem Memory Express () für kurzlebige Daten. NVMe
- R5— Speicheroptimierte Instances, die bei gleichen Instance-Kosten eine um bis zu 100% bessere Leistung als R4-Instances bieten.
- R4— Frühere Generation speicheroptimierter Instances.
- T4G—Kostengünstiger Allzweck-Instance-Typ der neuesten Generation, der auf ARM-basierten AWS Graviton2-Prozessoren basiert und eine grundlegende CPU-Leistung bietet, ein um bis zu 35% besseres Preis-Leistungs-Verhältnis als T3-Instances bietet und sich ideal für die Ausführung von Anwendungen mit moderater CPU-Auslastung eignet, bei denen es zu vorübergehenden Auslastungsspitzen kommt.
- T3— Kostengünstiger Burstable-Instance-Typ für allgemeine Zwecke, der eine grundlegende CPU-Leistung bietet und die Möglichkeit bietet, die CPU-Auslastung jederzeit und so lange wie nötig zu erhöhen.

Detaillierte Angaben zu den Instance-Klassen finden Sie unter Spezifikationen der Instanzklasse.

Eine bestimmte Instance-Klasse kann in einer bestimmten Region unterstützt werden oder nicht. Die folgende Tabelle gibt an, welche Instance-Klassen von Amazon DocumentDB in jeder Region unterstützt werden.

Unterstützte Instance-Klassen nach Region

Region	R6GD	R6G	R5	R4	T4G	T3
USA Ost (Ohio)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA Ost (Nord-Virginia)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
USA West (Oregon)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Afrika (Kapstadt)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Südamerika (São Paulo)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hongkong)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Hyderabad)			Unterstützt			Unterstützt
Asien-Pazifik (Mumbai)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Seoul)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Sydney)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt

Region	R6GD	R6G	R5	R4	T4G	T3
Asien-Pazifik (Singapur)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Asien-Pazifik (Tokio)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Kanada (Zentral)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Frankfurt)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Irland)	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt	Unterstützt
Europa (London)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Milan)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Paris)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Spain)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Europa (Stockholm)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
Naher Osten (VAE)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
China (Peking)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
China (Ningxia)		Unterstützt	Unterstützt		Unterstützt	Unterstützt

Region	R6GD	R6G	R5	R4	T4G	T3
Israel (Tel Aviv)		Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (US-West)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt
AWS GovCloud (US-Ost)	Unterstützt	Unterstützt	Unterstützt		Unterstützt	Unterstützt

Spezifikationen der Instanzklasse

Die folgende Tabelle enthält Einzelheiten zu den Amazon DocumentDB DocumentDB-Instance-Klassen, einschließlich der Instance-Typen, die in den einzelnen Klassen unterstützt werden. Erläuterungen zu den einzelnen Tabellenspalten finden Sie unterhalb der Tabelle.

Instance-Klasse	vCPU[1]	Speicher (GiB) [2]	NVMe Mehrstufiger SSD-Cache (GiB) 3	Max. Speichertemperatur (GiB) 4	Basis-/Burst-Bandbreite (Gbit/s) 5	Unterstützende Engines 6
R6G — Speicheroptimierte Instance-Klasse der aktuellen Generation, die auf Graviton2 basiert						
`db.r6g.large`	2	16	-	32	0,75/10	4.0.0 und 5.0.0
`db.r6g.xlarge`	4	32	-	63	1,25/10	4.0.0 und 5.0.0
`db.r6g.2xlarge`	8	64	-	126	2.5/10	4.0.0 und 5.0.0
`db.r6g.4xlarge`	16	128	-	252	5.0/10	4.0.0 und 5.0.0

Instance-Klasse	vCPU[1]	Speicher (GiB) [2]	NVMe Mehrstufiger SSD-Cache (GiB) 3	Max. Speichert emperatur (GiB) 4	Basis-/Burst-Bandbreite (Gbit/s) 5	Unterstützende Engines 6
db.r6g.8xlarge	32	256	-	504	12	4.0.0 und 5.0.0
db.r6g.12xlarge	48	384	-	756	20	4.0.0 und 5.0.0
db.r6g.16xlarge	64	512	-	1008	25	4.0.0 und 5.0.0
R6GD — Von der aktuellen Generation NVMe unterstützte Instance-Klasse, die auf Graviton2 basiert						
db.r6gd.xlarge	4	32	173	64	1,25/10	nur 5.0.0
db.r6gd.2xlarge	8	64	346	128	2,5/10	nur 5.0.0
db.r6gd.4xlarge	16	128	694	256	5,0/10	nur 5.0.0
db.r6gd.8xlarge	32	256	1388	512	12	nur 5.0.0
db.r6gd.12xlarge	48	384	2082	768	20	nur 5.0.0
db.r6gd.16xlarge	64	512	276	1024	25	nur 5.0.0
R5 — Speicheroptimierte Instance-Klasse der vorherigen Generation						
db.r5.large	2	16	-	31	0,75/10	3.6.0, 4.0.0 und 5.0.0

Instance-Klasse	vCPU[1]	Speicher (GiB) [2]	NVMe Mehrstufiger SSD-Cache (GiB) 3	Max. Speichertemperatur (GiB) 4	Basis-/Burst-Bandbreite (Gbit/s) 5	Unterstützende Engines 6
db.r5.xlarge	4	32	-	62	1,25/10	3.6.0, 4.0.0 und 5.0.0
db.r5.2xlarge	8	64	-	124	2.5/10	3.6.0, 4.0.0 und 5.0.0
db.r5.4xlarge	16	128	-	249	5.0/10	3.6.0, 4.0.0 und 5.0.0
db.r5.8xlarge	32	256	-	504	10	3.6.0, 4.0.0 und 5.0.0
db.r5.12xlarge	48	384	-	748	12	3.6.0, 4.0.0 und 5.0.0
db.r5.16xlarge	64	512	-	1008	20	3.6.0, 4.0.0 und 5.0.0
db.r5.24xlarge	96	768	-	1500	25	3.6.0, 4.0.0 und 5.0.0
R4 — Speicheroptimierte Instance-Klasse der vorherigen Generation						
db.r4.large	2	15,25	-	30	0,75/10	nur 3.6.0
db.r4.xlarge	4	30,5	-	60	1,25/10	nur 3.6.0
db.r4.2xlarge	8	61	-	120	2,5/10	nur 3.6.0
db.r4.4xlarge	16	122	-	240	5,0 /10	nur 3.6.0

Instance-Klasse	vCPU[1]	Speicher (GiB) [2]	NVMe Mehrstufiger SSD-Cache (GiB) 3	Max. Speichert emperatur (GiB) 4	Basis-/Burst-Bandbreite (Gbit/s) 5	Unterstützende Engines 6
`db.r4.8xlarge`	32	244	-	480	10	nur 3.6.0
`db.r4.16xlarge`	64	488	-	960	25	nur 3.6.0
T4G — Instance-Klassen mit Burstable Performance der neuesten Generation, die auf Graviton2 basieren						
`db.t4g.medium`	2	4	-	8.13	0,256/5	4.0.0 und 5.0.0
T3 — Instance-Klassen mit Burstable Performance der vorherigen Generation						
`db.t3.medium`	2	4	-	7,5	0,256/5	3.6.0, 4.0.0 und 5.0.0

Instance-Klasse	vCPU[1]	Speicher (GiB) [2]	NVMe Mehrstufiger SSD-Cache (GiB) 3	Max. Speichert emperatur (GiB) 4	Basis-/Burst-Bandbreite (Gbit/s) 5	Unterstützende Engines 6

1. vCPU — Die Anzahl der virtuellen Zentraleinheiten (CPUs). Eine virtuelle CPU ist eine Kapazitätseinheit, mit der Sie Instance-Klassen vergleichen können. Anstatt einen bestimmte n Prozessor für mehrere Monate oder Jahre zu erwerben oder zu leasen, wird jetzt Kapazität stundenweise gemietet. Unser Ziel besteht darin, unabhängig von der tatsächlich zu Grunde liegenden Hardware eine gleichbleibende Menge an CPU-Kapazität zu bieten.
2. Arbeitsspeicher (GiB) — Der RAM in Gigabyte, der der Instance zugewiesen ist. Häufig ist das Verhältnis zwischen Arbeitsspeicher- und vCPU konsistent.
3. NVMe Mehrstufiger SSD-Cache — Der Speicherplatz auf dem SSD-Volume, gemessen in Gigabyte, der als erweiterter Cache zum Speichern kurzlebiger Daten zugewiesen wird. Dieser Cache ist nur in Instanzen mit -Unterstützung verfügbar. NVMe
4. Max. Temp. Storage (GiB) — Der Speicherplatz, gemessen in Gigabyte, der der Instanz für nicht persistenten temporären Dateispeicher zugewiesen wurde. Bei Instances mit NVMe - Unterstützung wird dieser Speicher auf einem SSD-Volume gehostet. NVMe In allen anderen Fällen wird es im Amazon Elastic Block Store (EBS) gehostet.
5. Baseline-/Burst-Bandbreite (Gbit/s) — Die Burst-Bandbreite stellt die maximale Bandbreite in Gigabit pro Sekunde dar. Teilen Sie durch 8, um den erwarteten Durchsatz in Gigabyte pro Sekunde zu erhalten. Instanzen der Größe 4xlarge und kleiner haben eine Basisbandbreite. Um den zusätzlichen Bedarf zu decken, können sie einen I/O Netzwerk-Kreditmechanismus verwenden, um ihre Basisbandbreite zu überschreiten. Instances können Aufteilungsbandbreite für einen begrenzten Zeitraum verwenden, in der Regel zwischen 5 und 60 Minuten, abhängig von der Instancegröße.
6. Unterstützende Engines — Die Amazon DocumentDB DocumentDB-Engines, die die Instance-Klasse unterstützen.

NVMe-gestützte Instances

NVMe-gestützte Instances bieten eine bis zu 7-mal schnellere Abfrageleistung für Anwendungen mit großen Datensätzen, die den Arbeitsspeicher einer regulären Instanz überschreiten. Diese Instances nutzen den lokalen Solid-State-Drive-Speicher (SSDNVMe) auf der Basis von Non-Volatile Memory Express (), der auf r6gd-Instances verfügbar ist, um kurzlebige Daten zu speichern,

den netzwerkbasierten Speicherzugriff zu reduzieren und die Leselatenz und den Durchsatz zu verbessern.

Der lokale SSD-Speicherplatz ist in zwei Abschnitte unterteilt:

- Mehrstufiger Cache — Ungefähr 73% der lokalen SSD sind als Datenbank-Cache zugewiesen, sodass das System bis zu fünfmal mehr Datenbankseiten speichern kann als der Hauptspeicher allein. Die lokale SSD fungiert als Cache der zweiten Ebene, während der vorhandene In-Memory-Buffer-Cache der Cache der ersten Ebene bleibt. Die Abfrage greift nur dann auf externen Speicher zu, wenn sowohl im Puffer-Cache als auch im SSD-Cache ein Fehler auftritt.
- Temporärer Speicher — Die restlichen 27% sind für nicht persistenten temporären Dateispeicher reserviert, der für komplexe Abfragen mit Sortieroperationen oder ressourcenintensiven Vorgängen wie Indexerstellungen verwendet wird. Bei regulären Instances befindet sich der temporäre Speicherplatz auf einem Amazon Elastic Block Store (EBS) -Volume. Der lokal gehostete temporäre Speicher auf der SSD reduziert die Abfragelatenz bei Sortierungen um das bis zu Zweifache und beschleunigt ressourcenintensive Operationen wie Indexerstellungen.

Einzelheiten zum Typ der NVMe unterstützten Instanzen und zu deren Spezifikationen finden Sie unter. Spezifikationen der Instanzklasse

Themen

- Empfohlene Anwendungsfälle für NVMe -gestützte Instances
- Verwenden von NVMe -gestützten Instances mit Amazon DocumentDB
- Instances, die von der Überwachung unterstützt werden NVMe

Empfohlene Anwendungsfälle für NVMe -gestützte Instances

Wir empfehlen Ihnen, in den folgenden Szenarien Instances mit NVMe -backed zu verwenden:

- Leseintensive Workloads — Wenn Ihre Arbeitslast leseintensiv ist und Ihr Datensatz größer als der Puffer-Cache ist, was durch niedrige `BufferCacheHitRatio` und hohe `ReadIOPS` Messwerte gekennzeichnet ist, können NVMe -Instances Leistungsvorteile bieten.
- Aktualisierungsintensive Workloads — Wenn Ihre Arbeitslast aktualisierungsintensiv ist und die Garbage-Collection aufgrund der Leselatenz im Netzwerkspeicher nicht Schritt halten kann, können Instances, die auf dem Netzwerkspeicher basieren, helfen, das Problem zu beheben. NVMe

NVMe-gestützte Instances können für verschiedene Anwendungsfälle von Vorteil sein, darunter:

- Anwendungen auf Internetebene — Anwendungen wie Zahlungsabwicklung, Rechnungsstellung und E-Commerce, für die strenge Service Level Agreements (SLAs) gelten, können die Leistungsvorteile von NVMe Instances nutzen, die auf dem System basieren.
- Dashboards für Berichte in Echtzeit — Dashboards, die Hunderte von Abfragen zur Erfassung von Metriken und Daten ausführen, können von der geringen Latenz und dem hohen Durchsatz unterstützter Instanzen profitieren. NVMe
- Generative KI-Anwendungen — Anwendungen, die die Vektorsuche verwenden, um anhand von Millionen von Vektor-Einbettungen exakte oder nächstgelegene Nachbarn zu finden, können die hohe Leistung von -gestützten Instances nutzen. NVMe

Verwenden von NVMe -gestützten Instances mit Amazon DocumentDB

Um Instanzen von Amazon NVMe DocumentDB zu verwenden, gehen Sie wie folgt vor:

- Erstellen Sie einen Amazon DocumentDB-Cluster und fügen Sie eine der NVMe -gestützten Instance-Klassen hinzu. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster erstellen.
- Sie können auch einen vorhandenen Amazon DocumentDB-Cluster so ändern, dass er eine der Instance-Klassen mit NVMe -backed verwendet. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Informationen zur Überprüfung der Verfügbarkeit von Instances mit NVMe -Unterstützung in verschiedenen AWS Regionen finden Sie unter. Unterstützte Instance-Klassen nach Region

Wenn Sie von einer NVMe -gestützten Instance zu einer regulären Instance zurückkehren möchten, ändern Sie die Datenbank-Instance-Klasse Ihrer Amazon DocumentDB DocumentDB-Instance in eine ähnliche Instance-Klasse ohne Speicher. NVMe Wenn Ihre aktuelle Instance-Klasse beispielsweise 'db.r6gd.4xlarge' ist, wählen Sie 'db.r6g.4xlarge', um zurückzuschalten. Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

Instances, die von der Überwachung unterstützt werden NVMe

Zusätzlich zu den regulären Instance-Metriken, die bei Amazon verfügbar sind CloudWatch, geben NVMe -gestützte Instances zusätzliche Metriken aus, die sich NVMe auf SSD-basierten Speicher, IOPS und Durchsatz beziehen.

```
NVMeStorageCacheHitRatio
FreeNVMeStorage
ReadIOPSNVMeStorage
ReadLatencyNVMeStorage
ReadThroughputNVMeStorage
WriteIOPSNVMeStorage
WriteLatencyNVMeStorage
WriteThroughputNVMeStorage
```

Weitere Informationen zu diesen Metriken finden Sie unter NVMe-unterstützte Instance-Metriken

Amazon DocumentDB-Subnetzgruppen verwalten

Eine Virtual Private Cloud (VPC) ist ein virtuelles Netzwerk, das Ihrem AWS-Konto gewidmet ist. Es ist logisch von anderen virtuellen Netzwerken in der AWS Cloud isoliert. Sie können Ihre AWS Ressourcen, wie Amazon DocumentDB-Cluster, in Ihrer Amazon VPC starten. Sie können einen IP-Adressbereich für die VPC angeben, Subnetze hinzufügen, Sicherheitsgruppen zuordnen und Routing-Tabellen konfigurieren.

Ein Subnetz ist ein Bereich von IP-Adressen in Ihrer Amazon VPC. Sie können AWS Ressourcen in einem bestimmten Subnetz starten. Verwenden Sie öffentliche Subnetze für Ressourcen, die mit dem Internet verbunden sein müssen. Verwenden Sie private Subnetze für Ressourcen, die nicht mit dem Internet verbunden sein werden. Weitere Informationen zu öffentlichen und privaten Subnetzen finden Sie unter VPC and Subnet Basics im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Eine DB-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie in einer VPC erstellen, welche Sie dann für Ihre Cluster festlegen. Mithilfe einer Subnetzgruppe können Sie beim Erstellen von Clustern eine bestimmte VPC festlegen. Wenn Sie die `default` Subnetzgruppe verwenden, umfasst diese alle Subnetze in der VPC.

Jede DB-Subnetzgruppe sollte über Subnetze in mindestens zwei Availability Zones in einer bestimmten -Region verfügen. Wenn Sie einen DB-Cluster in einer VPC erstellen, müssen Sie eine DB-Subnetzgruppe auswählen. Amazon DocumentDB verwendet diese DB-Subnetzgruppe und Ihre bevorzugte Availability Zone, um ein Subnetz und eine IP-Adresse innerhalb dieses Subnetzes auszuwählen, die mit Ihrem Cluster verknüpft werden sollen. Wenn die primäre Instance ausfällt, kann Amazon DocumentDB eine entsprechende Replikat-Instance zur neuen primären Instance heraufstufen. Anschließend kann mithilfe einer IP-Adresse aus dem Subnetz, in dem sich der vorherige primäre Cluster befand, eine neue Replikat-Instance erstellt werden.

Wenn Amazon DocumentDB eine Instance in einer VPC erstellt, weist es Ihrem Cluster mithilfe einer IP-Adresse, die Sie aus Ihrer DB-Subnetzgruppe ausgewählt haben, eine Netzwerkschnittstelle zu. Es wird ausdrücklich empfohlen, den DNS-Namen zu verwenden, da sich die zugrunde liegende IP-Adresse ändern kann (z. B während eines Failovers). Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Endpunkte.

Informationen zum Erstellen Ihrer eigenen VPC und Subnetze finden Sie unter Arbeiten mit VPCs und Subnetzen im Amazon Virtual Private Cloud Cloud-Benutzerhandbuch.

Themen

- Eine Amazon DocumentDB-Subnetzgruppe erstellen
- Beschreibung einer Amazon DocumentDB-Subnetzgruppe
- Ändern einer Amazon DocumentDB-Subnetzgruppe
- Löschen einer Amazon DocumentDB-Subnetzgruppe

Eine Amazon DocumentDB-Subnetzgruppe erstellen

Wenn Sie einen Amazon DocumentDB-Cluster erstellen, müssen Sie eine Amazon VPC und die entsprechende Subnetzgruppe innerhalb dieser Amazon VPC auswählen, um Ihren Cluster zu starten. Subnetze bestimmen die Availability Zone und den IP-Bereich innerhalb der Availability Zone, die Sie zum Starten einer Instance verwenden möchten.

Eine Subnetzgruppe ist eine benannte Gruppe von Subnetzen (oder AZs), mit der Sie die Availability Zones angeben können, die Sie für den Start von Amazon DocumentDB DocumentDB-Instances verwenden möchten. In einem Cluster mit drei Instances wird beispielsweise empfohlen, dass jede dieser Instances separat bereitgestellt wird. AZs Dadurch wird die Hochverfügbarkeit optimiert. Wenn also eine einzelne AZ ausfällt, wirkt sich dies nur auf eine einzelne Instanz aus.

Derzeit können Amazon DocumentDB DocumentDB-Instances in bis zu drei Instanzen bereitgestellt werden. AZs Selbst wenn eine Subnetzgruppe mehr als drei Subnetze hat, können Sie nur drei dieser Subnetze verwenden, um einen Amazon DocumentDB-Cluster zu erstellen. Daher empfehlen wir, bei der Erstellung einer Subnetzgruppe nur die drei Subnetze auszuwählen, für die Sie Ihre Instances bereitstellen möchten.

Beispiel: Ein Cluster wird erstellt und Amazon DocumentDB wählt AZs {1A, 1B und 1C}. Wenn Sie versuchen, eine Instance in AZ {1D} zu erstellen, schlägt der API-Aufruf fehl. Wenn Sie sich jedoch dafür entscheiden, eine Instance zu erstellen, ohne die jeweilige AZ anzugeben, wählt Amazon DocumentDB in Ihrem Namen eine AZ aus. Amazon DocumentDB verwendet einen Algorithmus für

den Lastenausgleich zwischen den Instanzen AZs , um Ihnen zu helfen, eine hohe Verfügbarkeit zu erreichen. Wenn drei Instances bereitgestellt werden, werden sie standardmäßig über drei bereitgestellt AZs und nicht alle in einer einzigen AZ bereitgestellt.

Bewährte Methoden

- Wenn Sie keinen speziellen Grund haben, legen Sie immer eine Subnetzgruppe mit drei Subnetzen an. Dadurch wird sichergestellt, dass Cluster mit drei oder mehr Instances eine höhere Verfügbarkeit erreichen können, da Instances über drei Instanzen bereitgestellt werden. AZs
- Verteilen Sie Instanzen immer auf mehrere AZs , um eine hohe Verfügbarkeit zu erreichen. Platzieren Sie niemals alle Instances für einen Cluster in einer einzigen AZ.
- Da Failover-Ereignisse jederzeit auftreten können, sollten Sie nicht davon ausgehen, dass sich eine primäre Instance oder Replikat-Instance immer in einer bestimmten AZ befinden.

Wie erstellt man eine Subnetzgruppe

Sie können das AWS Management Console oder verwenden AWS CLI , um eine Amazon DocumentDB-Subnetzgruppe zu erstellen:

Using the AWS Management Console

Gehen Sie wie folgt vor, um eine Amazon DocumentDB-Subnetzgruppe zu erstellen.

So erstellen Sie eine Amazon DocumentDB-Subnetzgruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnet groups (Subnetzgruppen) und anschließend Create (Erstellen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Auf der Seite Create subnet group (Subnetzgruppe erstellen):

a. Im Abschnitt Subnet group details (Subnetzgruppen-Details:

   i. Name — Geben Sie einen aussagekräftigen Namen für die Subnetzgruppe ein.

   ii. Beschreibung – Geben Sie eine Beschreibung für die Subnetzgruppe ein.

b. Im Abschnitt Add subnets (Subnetze hinzufügen):

   i. VPC — Wählen Sie in der Liste eine VPC für diese Subnetzgruppe aus.

   ii. Führen Sie eine der folgenden Aktionen aus:

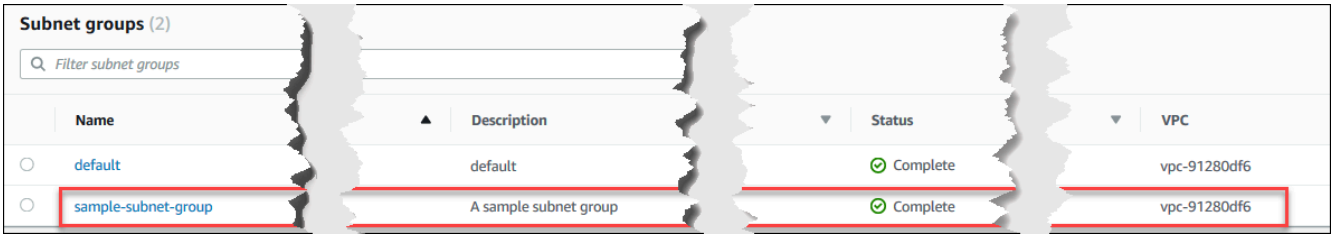
   - Um alle Subnetze in der ausgewählten VPC miteinzuschließen, wählen Sie Add all the subnets related to this VPC (Alle zu dieser VPC gehörenden Subnetze hinzufügen).
   - Um Subnetze für diese Subnetzgruppe festzulegen, führen Sie folgende Aufgaben für jede Availability Zone aus, für die Sie Subnetze miteinschließen möchten. Sie müssen mindestens zwei Availability Zones einschließen.

     A. Availability Zone — Wählen Sie in der Liste eine Availability Zone aus.

     B. Subnetz — Wählen Sie in der Liste ein Subnetz aus der ausgewählten Availability Zone für diese Subnetzgruppe aus.

     C. Wählen Sie Add subnet (Subnetz hinzufügen) aus.

4. Wählen Sie Create (Erstellen) aus. Wenn die Subnetzgruppe erstellt wurde, wird sie mit Ihren anderen Subnetzgruppen aufgeführt.



Using the AWS CLI

Bevor Sie mithilfe von eine Subnetzgruppe erstellen können, müssen Sie zunächst ermitteln AWS CLI, welche Subnetze verfügbar sind. Führen Sie den folgenden AWS CLI Vorgang aus, um die Availability Zones und ihre Subnetze aufzulisten.

Parameter:

- **`--db-subnet-group`**— Fakultativ. Bei Angabe einer bestimmten Subnetzgruppe werden die Availability Zones und Subnetze für diese Gruppe aufgelistet. Durch Auslassen dieses Parameters werden Availability Zones und Subnetze für alle Ihre Subnetzgruppen aufgelistet. Bei Angabe der `default`-Subnetzgruppe werden alle Subnetze der VPC aufgelistet.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default \
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

Für Windows:

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name default ^
    --query 'DBSubnetGroups[*].[DBSubnetGroupName,Subnets[*].
[SubnetAvailabilityZone.Name,SubnetIdentifier]]'
```

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

```
[
    [
        "default",
        [
            [
                "us-east-1a",
                "subnet-4e26d263"
            ],
            [
                "us-east-1c",
                "subnet-afc329f4"
            ],
            [
                "us-east-1e",
                "subnet-b3806e8f"
            ],
            [
                "us-east-1d",
```

```
                "subnet-53ab3636"
            ],
            [
                "us-east-1b",
                "subnet-991cb8d0"
            ],
            [
                "us-east-1f",
                "subnet-29ab1025"
            ]
        ]
    ]
]
```

Sie können eine neue Subnetzgruppe erstellen, indem Sie die Ausgabe des vorherigen Vorgangs verwenden. Die neue Subnetzgruppe muss über Subnetze aus mindestens zwei Availability Zones verfügen.

Parameter:

- **`--db-subnet-group-name`**—Erforderlich. Der Name für diese Subnetzgruppe.
- **`--db-subnet-group-description`**—Erforderlich. Beschreibung dieser Subnetzgruppe.
- **`--subnet-ids`**—Erforderlich. Eine Liste der Subnetze in dieser Subnetzgruppe. Beispiel: `subnet-53ab3636`.
- --Schlagworte — Optional. Eine Liste der Tags (Schlüssel-Wert-Paare) zum Anfügen an diese Subnetzgruppe.

Der folgende Code erstellt die Subnetzgruppe `sample-subnet-group` mit drei Subnetzen, `subnet-4e26d263`, `subnet-afc329f4` und `subnet-b3806e8f`.

Für Linux, macOS oder Unix:

```
aws docdb create-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --db-subnet-group-description "A sample subnet group" \
    --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f \
    --tags Key=tag1,Value=One Key=tag2,Value=2
```

Für Windows:

```
aws docdb create-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group ^
    --db-subnet-group-description "A sample subnet group" ^
    --subnet-ids subnet-4e26d263 subnet-afc329f4 subnet-b3806e8f ^
    --tags Key=tag1,Value=One Key=tag2,Value=2
```

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

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupDescription": "A sample subnet group",
        "DBSubnetGroupName": "sample-subnet-group",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetIdentifier": "subnet-4e26d263",
                "SubnetStatus": "Active"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetIdentifier": "subnet-afc329f4",
                "SubnetStatus": "Active"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetIdentifier": "subnet-b3806e8f",
                "SubnetStatus": "Active"
            }
        ],
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "SubnetGroupStatus": "Complete"
    }
}
```

Beschreibung einer Amazon DocumentDB-Subnetzgruppe

Sie können das AWS Management Console oder das verwenden AWS CLI , um die Details einer Amazon DocumentDB-Subnetzgruppe abzurufen.

Using the AWS Management Console

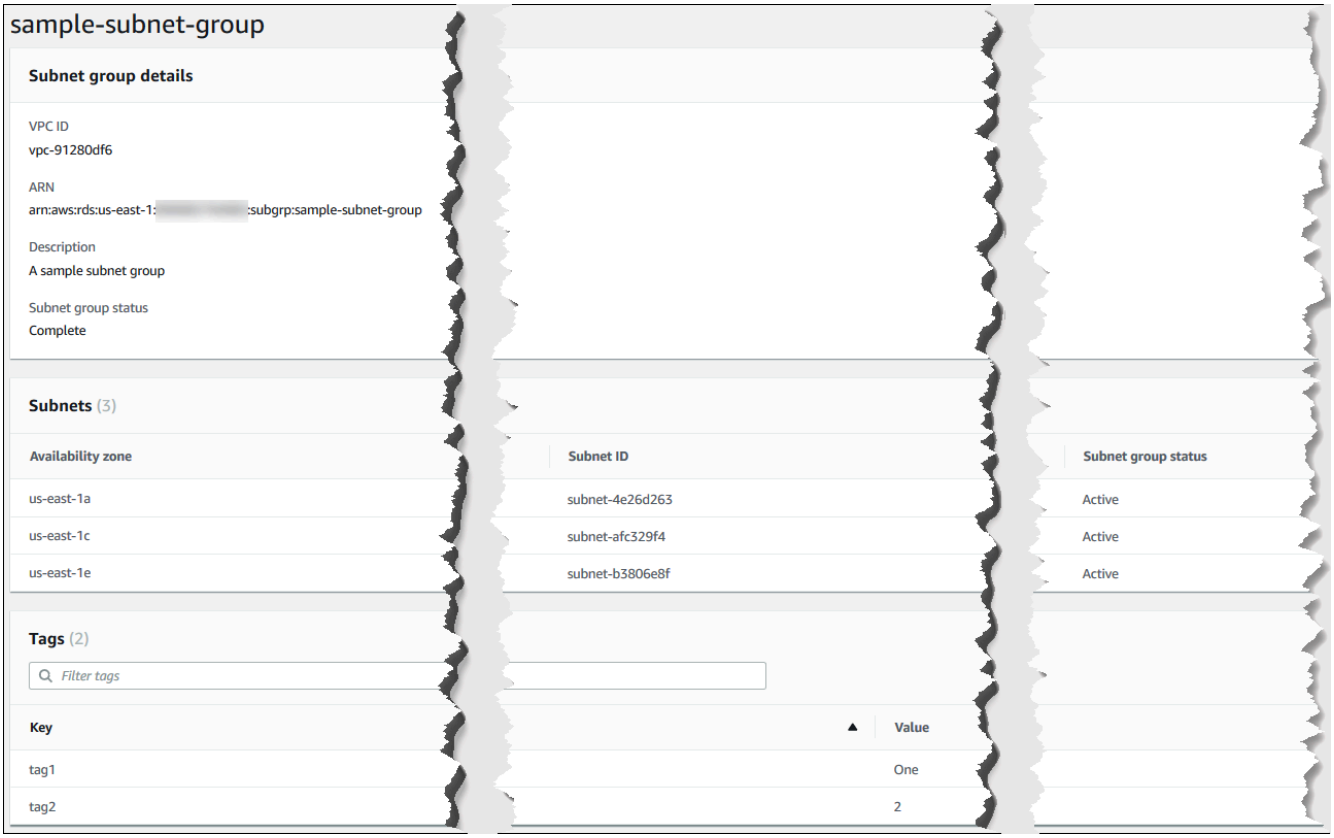
Das folgende Verfahren zeigt Ihnen, wie Sie die Details einer Amazon DocumentDB-Subnetzgruppe abrufen können.

So suchen Sie die Details einer Subnetzgruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Um die Details einer Subnetzgruppe anzuzeigen, wählen Sie den Namen der Subnetzgruppe.



Using the AWS CLI

Um die Details einer Amazon DocumentDB-Subnetzgruppe zu finden, verwenden Sie den `describe-db-subnet-groups` Vorgang mit dem folgenden Parameter.

Parameter

- `--db-subnet=group-name`— Optional. Falls angegeben, werden Details für die benannte Subnetzgruppe aufgelistet. Falls nicht angegeben, werden Details für bis zu 100 Subnetzgruppen aufgelistet.

Example

Der folgende Code listet die Details für die Subnetzgruppe `sample-subnet-group` auf, die im Abschnitt Eine Amazon DocumentDB-Subnetzgruppe erstellen erstellt wurde.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-subnet-groups \
```

```
    --db-subnet-group-name sample-subnet-group
```

Für Windows:

```
aws docdb describe-db-subnet-groups ^
    --db-subnet-group-name sample-subnet-group
```

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

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "VpcId": "vpc-91280df6",
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupName": "sample-subnet-group",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1a"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-4e26d263"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1c"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-afc329f4"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1e"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-b3806e8f"
            }
        ],
        "DBSubnetGroupDescription": "A sample subnet group"
    }
}
```

Ändern einer Amazon DocumentDB-Subnetzgruppe

Sie können das AWS Management Console oder verwenden AWS CLI , um die Beschreibung einer Subnetzgruppe zu ändern oder Subnetze zu einer Amazon DocumentDB-Subnetzgruppe hinzuzufügen oder zu entfernen. Sie können die `default`-Subnetzgruppe jedoch nicht ändern.

Using the AWS Management Console

Sie können das verwenden AWS Management Console , um die Beschreibung einer Subnetzgruppe zu ändern oder Subnetze hinzuzufügen oder zu entfernen. Denken Sie daran, dass beim Abschluss des Vorgangs mindestens zwei Availability Zones mit Ihrer Subnetzgruppe verbunden sein müssen.

So ändern Sie Ihre Subnetzgruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus. Klicken Sie dann auf die Schaltfläche links neben dem Namen der Subnetzgruppe. Denken Sie daran, dass Sie die `default`-Subnetzgruppe nicht ändern können.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus.
4. Beschreibung — Um die Beschreibung Ihrer Subnetzgruppe zu ändern, geben Sie eine neue Beschreibung ein.
5. Wenn Sie die mit Ihrer Subnetzgruppe verbundenen Subnetze ändern möchten, führen Sie im Abschnitt Add subnets (Subnetze hinzufügen) eine oder mehrere der folgenden Aktionen durch:
   - Um alle Subnetze aus dieser Subnetzgruppe zu entfernen, wählen Sie Remove all (Alle entfernen) aus.
   - Wenn Sie bestimmte Subnetze aus dieser Subnetzgruppe entfernen möchten, wählen Sie für jedes einzelne Subnetz, das Sie entfernen möchten, Remove (Entfernen) aus.

- Wenn Sie alle mit dieser VPC verbundenen Subnetze hinzufügen möchten, wählen Sie Add all the subnets related to this VPC (Alle zu dieser VPC gehörenden Subnetze hinzufügen) aus.
- Wenn Sie bestimmte Subnetze zu dieser Subnetzgruppe hinzufügen möchten, führen Sie die folgenden Aufgaben für jede Availability Zone aus, der Sie ein Subnetz hinzufügen möchten.

  a. Availability Zone — Wählen Sie in der Liste eine neue Availability Zone aus.

  b. Subnetz — Wählen Sie in der Liste ein Subnetz aus der ausgewählten Availability Zone für diese Subnetzgruppe aus.

  c. Wählen Sie Add subnet (Subnetz hinzufügen) aus.

6. Im Bestätigungsdialogfeld:
   - Wählen Sie Modify (Ändern), um diese Änderungen an der Subnetzgruppe vorzunehmen.
   - Wenn die Subnetzgruppe unverändert bleiben soll, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Sie können den verwenden AWS CLI , um die Beschreibung einer Subnetzgruppe zu ändern oder Subnetze hinzuzufügen oder zu entfernen. Denken Sie daran, dass beim Abschluss des Vorgangs mindestens zwei Availability Zones mit Ihrer Subnetzgruppe verbunden sein müssen. Es ist nicht möglich, die `default`- Subnetzgruppe zu ändern.

Parameter:

- `--db-subnet-group-name`—Erforderlich. Der Name der Amazon DocumentDB-Subnetzgruppe, die Sie ändern.
- `--subnet-ids`—Erforderlich. Eine Liste aller Subnetze, die nach dieser Änderung Teil der Subnetzgruppe sein sollen.

> **Important**
>
> Alle Subnetze, die sich derzeit in der Subnetzgruppe befinden und die nicht in dieser Liste aufgeführt sind, werden aus der Subnetzgruppe entfernt. Wenn Sie Subnetze, die sich derzeit in der Subnetzgruppe befinden, beibehalten möchten, müssen Sie diese in die Liste aufnehmen.

- `--db-subnet-group-description`— Fakultativ. Beschreibung der Subnetzgruppe.

Example

Der folgende Code ändert die Beschreibung und ersetzt die vorhandenen Subnetze durch die Subnetze `subnet-991cb8d0`, `subnet-53ab3636` und `subnet-29ab1025`.

Für Linux, macOS oder Unix:

```
aws docdb modify-db-subnet-group \
    --db-subnet-group-name sample-subnet-group \
    --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 \
    --db-subnet-group-description "Modified subnet group"
```

Für Windows:

```
aws docdb modify-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group ^
    --subnet-ids subnet-991cb8d0 subnet-53ab3636 subnet-29ab1025 ^
    --db-subnet-group-description "Modified subnet group"
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus (JSON-Format). Beachten Sie, dass dies die gleiche Subnetzgruppe ist, die im Abschnitt Eine Amazon DocumentDB-Subnetzgruppe erstellen erstellt wurde. Die Subnetze in der Subnetzgruppe werden jedoch durch die in der Operation`modify-db-subnet-group` angegebenen ersetzt.

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupArn": "arn:aws:rds:us-east-1:123SAMPLE012:subgrp:sample-
subnet-group",
        "DBSubnetGroupDescription": "Modified subnet group",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1d"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-53ab3636"
            },
```

```
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1b"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-991cb8d0"
            },
            {
                "SubnetAvailabilityZone": {
                    "Name": "us-east-1f"
                },
                "SubnetStatus": "Active",
                "SubnetIdentifier": "subnet-29ab1025"
            }
        ],
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupName": "sample-subnet-group"
    }
}
```

Löschen einer Amazon DocumentDB-Subnetzgruppe

Sie können das AWS Management Console oder verwenden AWS CLI , um eine Amazon DocumentDB-Subnetzgruppe zu löschen. Sie können jedoch nicht die `default`-Subnetzgruppe löschen.

Using the AWS Management Console

Sie können den verwenden AWS Management Console , um eine Subnetzgruppe zu löschen. Die `default`-Subnetzgruppe kann jedoch nicht gelöscht werden.

So löschen Sie eine Subnetzgruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus. Klicken Sie dann auf die Schaltfläche links neben dem Namen der Subnetzgruppe. Denken Sie daran, dass Sie die `default`-Subnetzgruppe nicht löschen können.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie Aktionen und anschließend Löschen aus.
4. Im Bestätigungsdialogfeld:
   - Wählen Sie Delete (Löschen), um die Subnetzgruppe zu löschen.
   - Wählen Sie Cancel (Abbrechen), um die Subnetzgruppe zu erhalten.

Using the AWS CLI

Um eine Amazon DocumentDB-Subnetzgruppe mithilfe von zu löschen AWS CLI, verwenden Sie den `delete-db-subnet-group` Vorgang mit dem folgenden Parameter.

Parameter

- `--db-subnet-group-name`—Erforderlich. Der Name der Amazon DocumentDB-Subnetzgruppe, die gelöscht werden soll. Denken Sie daran, dass Sie die `default`-Subnetzgruppe nicht löschen können.

Example

Der folgende Code löscht `sample-subnet-group`.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

Für Windows:

```
aws docdb delete-db-subnet-group ^
    --db-subnet-group-name sample-subnet-group
```

Diese Operation erzeugt keine Ausgabe.

Amazon DocumentDB Hochverfügbarkeit und Replikation

Sie können Hochverfügbarkeit und Leseskalierung in Amazon DocumentDB (mit MongoDB-Kompatibilität) erreichen, indem Sie Replica-Instances verwenden. Ein einzelner Amazon DocumentDB-Cluster unterstützt eine einzelne primäre Instance und bis zu 15 Replikat-Instances. Diese Instances können über mehrere Availability Zones innerhalb der Cluster-Region verteilt werden. Die primäre Instance akzeptiert Lese- und Schreibverkehr, und Replikat-Instances akzeptieren nur Leseanforderungen.

Das Cluster-Volume besteht aus mehreren Kopien der Daten für den Cluster. Die Daten im Cluster-Volume werden jedoch als einzelnes logisches Volume für die primäre Instance und für Amazon DocumentDB DocumentDB-Replikate im Cluster dargestellt. Replikat-Instances sind „Eventually Consistent“. Sie geben Abfrageergebnisse mit minimaler Replikatverzögerung zurück – im Normalfall beträgt die Verzögerung weniger als 100 Millisekunden, nachdem die primäre Instance eine Aktualisierung geschrieben hat. Die Replica-Verzögerung variiert in Abhängigkeit vom Veränderungsgrad in der Datenbank. Das heißt, in Zeiten, in denen eine große Anzahl von Schreiboperationen für die Datenbank durchgeführt wird, kann es zu einer Erhöhung der Replikationsverzögerung kommen.

Skalierung von Lesevorgängen

Amazon DocumentDB DocumentDB-Replikate eignen sich gut für die Leseskalierung, da sie ausschließlich für Lesevorgänge auf Ihrem Cluster-Volume vorgesehen sind. Schreibvorgänge werden von der primären Instance verwaltet. Das Cluster-Volume wird von allen Instances in Ihrem Cluster gemeinsam genutzt. Daher müssen Sie nicht für jedes Amazon DocumentDB DocumentDB-Replikat eine Kopie der Daten replizieren und verwalten.

Hohe Verfügbarkeit

Wenn Sie einen Amazon DocumentDB-Cluster erstellen, stellt Amazon DocumentDB abhängig von der Anzahl der Availability Zones in der Subnetzgruppe (es müssen mindestens zwei vorhanden sein) Instances in den Availability Zones bereit. Wenn Sie Instances im Cluster erstellen, verteilt Amazon DocumentDB die Instances automatisch auf die Availability Zones in einer Subnetzgruppe, um den Cluster auszugleichen. Diese Vorgehensweise verhindert auch, dass sich alle Instances in derselben Availability Zone befinden.

Beispiel

Stellen Sie sich zur Veranschaulichung ein Beispiel vor, in dem Sie einen Cluster erstellen, der über eine Subnetzgruppe mit drei Availability Zones verfügt: AZ1, und. AZ2AZ3

Wenn die erste Instance im Cluster erstellt wird, ist diese die primäre Instance und befindet sich in einer der Availability Zones. In diesem Beispiel ist es in AZ1. Die zweite erstellte Instanz ist eine Replikatinstanz und befindet sich beispielsweise AZ2in einer der beiden anderen Availability Zones. Die dritte erstellte Instanz ist eine Replikatinstanz und befindet sich in der verbleibenden Availability Zone,. AZ3 Wenn Sie mehrere Instances erstellen, werden diese über die Availability Zones verteilt, sodass der Cluster ausbalanciert ist.

Wenn in der primären Instanz (AZ1) ein Fehler auftritt, wird ein Failover ausgelöst und eines der vorhandenen Replikate wird zur primären Instanz heraufgestuft. Wenn die alte Primäranlage wiederhergestellt wird, wird sie zu einem Replikat in derselben Availability Zone, in der sie bereitgestellt wurde (). AZ1 Wenn Sie einen Drei-Instanz-Cluster bereitstellen, behält Amazon DocumentDB diesen Drei-Instanz-Cluster weiterhin bei. Amazon DocumentDB kümmert sich automatisch um die Erkennung, den Failover und die Wiederherstellung von Instance-Ausfällen ohne manuelles Eingreifen.

Wenn Amazon DocumentDB einen Failover durchführt und eine Instance wiederherstellt, verbleibt die wiederhergestellte Instance in der Availability Zone, in der sie ursprünglich bereitgestellt wurde. Die Rolle der Instance ändert sich möglicherweise von der primären Instance zum Replikat. Dadurch wird ein Szenario verhindert, in dem eine Reihe von Failovers in allen Instances auftreten könnten, die sich in derselben Availability Zone befinden.

Sie können Amazon DocumentDB DocumentDB-Replikate als Failover-Ziele angeben. Das heißt, wenn die primäre Instance ausfällt, wird das angegebene Amazon DocumentDB DocumentDB-Replikat oder das Replikat aus einer Stufe zur primären Instance heraufgestuft. Es gibt dann eine kurze Unterbrechung, während der Lese- und Schreibanfragen an die primäre Instance mit einer Ausnahme fehlschlagen. Wenn Ihr Amazon DocumentDB-Cluster keine Amazon DocumentDB DocumentDB-Replikate enthält und die primäre Instance ausfällt, wird sie neu erstellt. Das Heraufstufen eines Amazon DocumentDB-Replikats ist viel schneller als das Neuerstellen der primären Instance.

Für Hochverfügbarkeitsszenarien empfehlen wir, ein oder mehrere Amazon DocumentDB DocumentDB-Replikate zu erstellen. Diese Replikate sollten derselben Instance-Klasse wie die primäre Instance angehören und sich in unterschiedlichen Availability Zones für Ihren Amazon DocumentDB-Cluster befinden.

Weitere Informationen finden Sie hier:

- Grundlegendes zur Amazon DocumentDB-Cluster-Fehlertoleranz
- Amazon DocumentDB-Failover
  - Steuerung des Failover-Ziels

Hohe Verfügbarkeit mit globalen Clustern

Um eine hohe Verfügbarkeit für mehrere zu gewährleisten AWS-Regionen, können Sie globale Amazon DocumentDB-Cluster einrichten. Jeder globale Cluster erstreckt sich über mehrere Regionen und ermöglicht globale Lesevorgänge mit geringer Latenz und die Notfallwiederherstellung nach Ausfällen in einer Region. AWS-Region Amazon DocumentDB übernimmt automatisch die Replikation aller Daten und Updates von der primären Region in jede der sekundären Regionen.

Hinzufügen von -Replicas

Die erste dem Cluster hinzugefügte Instance ist die primäre Instance. Jede Instance, die nach der ersten Instance hinzugefügt wird, ist eine Replikat-Instance. Ein Cluster kann zusätzlich zur primären bis zu 15 Replikatinstanzen haben.

Wenn Sie mit dem einen Cluster erstellen AWS Management Console, wird gleichzeitig automatisch eine primäre Instanz erstellt. Um ein Replikat gleichzeitig mit der Erstellung des Clusters und der primären Instance zu erstellen, wählen Sie Create replica in different zone (Replikat in unterschiedlicher Zone erstellen) aus. Weitere Informationen finden Sie in Schritt 4.d unter Einen Amazon DocumentDB-Cluster erstellen. Informationen zum Hinzufügen weiterer Replikate zu einem Amazon DocumentDB-Cluster finden Sie unter. Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster

Wenn Sie den AWS CLI zur Erstellung Ihres Clusters verwenden, müssen Sie Ihre Primär- und Replikat-Instances explizit erstellen. Weitere Informationen finden Sie in den folgenden Themen im Abschnitt AWS CLI„Verwenden von“:

- Einen Amazon DocumentDB-Cluster erstellen
- Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster

Amazon DocumentDB-Failover

In bestimmten Fällen, z. B. bei bestimmten Arten von geplanten Wartungsarbeiten oder in dem unwahrscheinlichen Fall, dass ein primärer Knoten oder eine Availability Zone ausfällt, erkennt Amazon DocumentDB (mit MongoDB-Kompatibilität) den Ausfall und ersetzt den primären Knoten. Während eines Failovers wird die Ausfallzeit für Schreibvorgänge minimiert. Das liegt daran, dass die Rolle des primären Knotens auf eine der Read Replicas übergeht, statt dass ein neuer primärer Knoten erstellt und bereitgestellt werden muss. Durch Ausfallerkennung und Replikatheraufstufung wird sichergestellt, dass Sie weiter in den neuen primären Knoten schreiben können, sobald die Heraufstufung abgeschlossen wurde.

Damit der Failover funktioniert, muss Ihr Cluster über mindestens zwei Instances verfügen — eine primäre und mindestens eine Replikat-Instance.

> **Note**
>
> Dieses Thema gilt nur für originale, auf Amazon DocumentDB DocumentDB-Instances basierende Cluster. Es gilt nicht für elastische oder globale Cluster.

Steuerung des Failover-Ziels

Amazon DocumentDB bietet Ihnen Failover-Stufen, mit denen Sie kontrollieren können, welche Replikat-Instance bei einem Failover zur primären Instanz heraufgestuft wird.

Failover-Stufen

Jede Replikat-Instance ist einer Failover-Stufe (0—15) zugeordnet. Wenn aufgrund von Wartungsarbeiten oder eines unwahrscheinlichen Hardwarefehlers ein Failover auftritt, wechselt die primäre Instanz zu einem Replikat mit der höchsten Priorität (der niedrigsten Stufe). Wenn mehrere Replikate dieselbe Prioritätsstufe haben, erfolgt ein Failover der primären Ebene auf das Replikat dieser Stufe, dessen Größe der vorherigen Primärstufe am nächsten kommt.

Indem Sie die Failover-Stufe für eine Gruppe ausgewählter Replikate auf `0` (höchste Priorität) setzen, können Sie sicherstellen, dass ein Failover auf eines der Replikate in dieser Gruppe wechselt. Sie können effektiv verhindern, dass bestimmte Replikate im Falle eines Failover zur primären Instance hochgestuft werden, indem Sie diesen Replikaten eine niedrige Stufe (hohe Anzahl) zuweisen. Dies ist nützlich, wenn bestimmte Replikate von einer Anwendung stark genutzt werden und ein Failover auf eine von ihnen eine kritische Anwendung negativ beeinflussen würde.

Sie können die Failover-Stufe einer Instance beim Erstellen oder später festlegen. Das Festlegen einer Instance-Failover-Stufe durch Ändern der Instance löst keinen Failover aus. Weitere Informationen finden Sie in den folgenden Themen:

- Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster
- Ändern einer Amazon DocumentDB DocumentDB-Instance

Wenn Sie einen Failover manuell einleiten, haben Sie zwei Möglichkeiten zur Steuerung, welche Replikat-Instance auf primär umgestellt wird: die Failover-Stufen wie zuvor beschrieben und den Parameter `--target-db-instance-identifier`.

--target-db-instance-identifier

Zum Testen können Sie mit der Operation `failover-db-cluster` ein Failover-Ereignis erzwingen. Mit dem Parameter `--target-db-instance-identifier` können Sie festlegen, welches Replikat auf primär umgestellt werden soll. Die Verwendung des Parameters `--target-db-instance-identifier` ersetzt die Failover-Prioritätsstufe. Wenn Sie den Parameter `--target-db-instance-identifier` nicht angeben, entspricht die Primär-Failover-Funktion der Failover-Prioritätsstufe.

Was passiert bei einem Failover

Das Failover wird automatisch von Amazon DocumentDB durchgeführt, sodass Ihre Anwendungen den Datenbankbetrieb so schnell wie möglich wieder aufnehmen können, ohne dass ein Administrator eingreifen muss.

- Wenn Sie beim Failover eine Amazon DocumentDB-Replikat-Instance in derselben oder einer anderen Availability Zone haben: Amazon DocumentDB dreht den Canonical Name Record (CNAME) für Ihre Instance um, sodass er auf das fehlerfreie Replikat verweist, das wiederum zum neuen primären Replikat heraufgestuft wird. Das Failover wird in der Regel innerhalb von 30 Sekunden vom Anfang bis zum Ende abgeschlossen.
- Wenn Sie keine Amazon DocumentDB-Replikatinstanz haben (z. B. einen einzelnen Instance-Cluster): Amazon DocumentDB versucht, eine neue Instance in derselben Availability Zone wie die ursprüngliche Instance zu erstellen. Dieser Austausch der ursprünglichen Instance wird nach bestem Bemühen durchgeführt, ist aber nicht immer erfolgreich, z. B. wenn ein Problem vorliegt, das sich allgemein auf die Availability Zone auswirkt.

Bei Verbindungsunterbrechung muss Ihre Anwendung versuchen, die Verbindung zur Datenbank wiederherzustellen.

Testen eines Failovers

Bei einem Failover für einen Cluster wird eine der Amazon DocumentDB DocumentDB-Repliken (schreibgeschützte Instances) im Cluster zur primären Instance (Cluster-Writer) heraufgestuft.

Wenn die primäre Instance ausfällt, wechselt Amazon DocumentDB automatisch zu einem Amazon DocumentDB DocumentDB-Replikat, falls eines existiert. Sie können ein Failover erzwingen, wenn Sie einen Ausfall einer primären Instance zum Testen simulieren möchten. Jede Instance in einem Cluster hat eine eigene Endpunkt-Adresse. Aus diesem Grund müssen Sie alle bestehenden Verbindungen, die diese Endpunktadressen verwenden, bereinigen und wiederherstellen, wenn der Failover abgeschlossen ist.

Um einen Failover zu erzwingen, verwenden Sie die Operation `failover-db-cluster` mit diesen Parametern.

- `--db-cluster-identifier`—Erforderlich. Der Name des Clusters, der einen Failover durchführen soll.
- `--target-db-instance-identifier`— Fakultativ. Der Name der Instance, die zur primären Instance befördert werden soll.

Example

Die folgende Operation erzwingt einen Failover des Clusters `sample-cluster`. Es wird nicht angegeben, welche Instance zur neuen primären Instance gemacht werden soll, sodass Amazon DocumentDB die Instance entsprechend der Priorität der Failover-Tier-Priorität auswählt.

Für Linux, macOS oder Unix:

```
aws docdb failover-db-cluster \
   --db-cluster-identifier sample-cluster
```

Für Windows:

```
aws docdb failover-db-cluster ^
   --db-cluster-identifier sample-cluster
```

Die folgende Operation erzwingt einen Failover des Clusters `sample-cluster` und legt fest, dass `sample-cluster-instance` in die primäre Rolle befördert werden soll. (Beachten Sie `"IsClusterWriter": true` in der Ausgabe.)

Für Linux, macOS oder Unix:

```
aws docdb failover-db-cluster \
   --db-cluster-identifier sample-cluster \
   --target-db-instance-identifier sample-cluster-instance
```

Für Windows:

```
aws docdb failover-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --target-db-instance-identifier sample-cluster-instance
```

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

```
{
    "DBCluster": {
        "HostedZoneId": "Z2SUY0A1719RZT",
        "Port": 27017,
        "EngineVersion": "3.6.0",
        "PreferredMaintenanceWindow": "thu:04:05-thu:04:35",
        "BackupRetentionPeriod": 1,
        "ClusterCreateTime": "2018-06-28T18:53:29.455Z",
        "AssociatedRoles": [],
        "DBSubnetGroup": "default",
        "MasterUsername": "master-user",
        "Engine": "docdb",
        "ReadReplicaIdentifiers": [],
        "EarliestRestorableTime": "2018-08-21T00:04:10.546Z",
        "DBClusterIdentifier": "sample-cluster",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-cluster-instance",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": true
            },
            {
```

```
                "DBInstanceIdentifier": "sample-cluster-instance-00",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": false
            },
            {
                "DBInstanceIdentifier": "sample-cluster-instance-01",
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1,
                "IsClusterWriter": false
            }
        ],
        "AvailabilityZones": [
            "us-east-1b",
            "us-east-1c",
            "us-east-1a"
        ],
        "DBClusterParameterGroup": "default.docdb3.6",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "IAMDatabaseAuthenticationEnabled": false,
        "AllocatedStorage": 1,
        "LatestRestorableTime": "2018-08-22T21:57:33.904Z",
        "PreferredBackupWindow": "00:00-00:30",
        "StorageEncrypted": false,
        "MultiAZ": true,
        "Status": "available",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-12345678"
            }
        ],
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    }
}
```

Replikationsverzögerung

Die Verzögerung bei der Replikation beträgt in der Regel 50 ms oder weniger. Die häufigsten Gründe für eine erhöhte Replikatverzögerung sind:

- Eine hohe Schreibrate auf der Primärseite, die dazu führt, dass die Lesereplikate hinter die Primärreplikate zurückfallen.
- Konflikt auf den Read Replicas zwischen Abfragen mit langer Laufzeit (z. B. umfangreiche sequentielle Scans, Aggregationsabfragen) und eingehender Schreibreplikation.
- Sehr große Anzahl gleichzeitiger Abfragen an den Read Replicas.

Um die Verzögerung bei der Replikation zu minimieren, sollten Sie die folgenden Methoden zur Problembehandlung ausprobieren:

- Wenn Sie eine hohe Schreibrate oder eine hohe CPU-Auslastung haben, empfehlen wir Ihnen, die Instances in Ihrem Cluster zu skalieren.
- Wenn Ihre Read Replicas lang andauern und die abgefragten Dokumente sehr häufig aktualisiert werden, sollten Sie in Erwägung ziehen, Ihre Abfragen mit langer Laufzeit zu ändern oder sie für das Primär-/Schreibreplikat auszuführen, um Konflikte bei den Read Replicas zu vermeiden.
- Wenn es nur bei den Read Replicas eine sehr große Anzahl gleichzeitiger Abfragen oder eine hohe CPU-Auslastung gibt, besteht eine weitere Möglichkeit darin, die Anzahl der Read Replicas zu skalieren, um die Arbeitslast zu verteilen.
- Da die Verzögerung bei der Replikation auf einen hohen Schreibdurchsatz und lange laufende Abfragen zurückzuführen ist, empfehlen wir, die Replikationsverzögerung mithilfe der DBCluster ReplicaLagMaximum CW-Metrik in Kombination mit dem langsamen Abfrage-Logger und den /-Metriken zu beheben. `WriteThroughput` `WriteIOPS`

Im Allgemeinen empfehlen wir, dass alle Ihre Replikate vom gleichen Instance-Typ sind, damit ein Cluster-Failover nicht zu Leistungseinbußen führt.

Wenn Sie zwischen Hochskalierung und Ausskalierung wählen (z. B. sechs kleinere Instances gegenüber drei größeren Instances), empfehlen wir generell, zuerst zu versuchen, vor dem Skalieren zu versuchen, bevor Sie die Skalierung verlassen, da Sie so einen größeren Puffer-Cache pro DB-Instance erhalten.

Proaktiv sollten Sie einen Alarm für die Verzögerung bei der Replikation einrichten und den Schwellenwert auf einen Wert setzen, der Ihrer Meinung nach die Obergrenze dafür darstellt, wie weit Ihre Daten auf Replikatinstanzen zurückliegen (oder „veraltet“) sein können, bevor sie die Funktionalität Ihrer Anwendung beeinträchtigen. Im Allgemeinen empfehlen wir, den Schwellenwert für die Replikationsverzögerung aufgrund vorübergehender Workloads für mehrere Datenpunkte zu überschreiten, bevor ein Alarm ausgelöst wird.

> **Note**
>
> Darüber hinaus empfehlen wir, einen weiteren Alarm für Replikationsverzögerungen von mehr als 10 Sekunden einzurichten. Wenn Sie diesen Schwellenwert für mehrere Datenpunkte überschreiten, empfehlen wir Ihnen, Ihre Instances zu skalieren oder Ihren Schreibdurchsatz auf der primären Instance zu reduzieren.

Amazon DocumentDB DocumentDB-Indizes verwalten

Themen

- Amazon DocumentDB DocumentDB-Indexerstellung
- Pflege Amazon DocumentDB DocumentDB-Indizes

Amazon DocumentDB DocumentDB-Indexerstellung

Für die Erstellung von Indizes in Amazon DocumentDB müssen eine Reihe von Entscheidungen getroffen werden:

- Wie schnell muss es abgeschlossen werden?
- Kann auf die Sammlung nicht zugegriffen werden, während der Build ausgeführt wird?
- Wie viel Rechenleistung einer Instanz kann dem Build zugewiesen werden?
- Welche Art von Index sollte erstellt werden?

Dieser Abschnitt hilft Ihnen bei der Beantwortung dieser Fragen und enthält die Befehle und Überwachungsbeispiele zum Erstellen eines Amazon DocumentDB DocumentDB-Indexes für Ihre instanzbasierte Clustersammlung.

Richtlinien

Die folgenden Richtlinien enthalten grundlegende Grenzwerte und Kompromisse bei der Konfiguration bei der Erstellung neuer Indizes:

- Unterstützung für Amazon DocumentDB DocumentDB-Versionen — Während die Single-Worker-Indizierung in allen Amazon DocumentDB DocumentDB-Versionen unterstützt wird, wird die

Multi-Worker-Indizierung nur in den Amazon DocumentDB DocumentDB-Versionen 4.0 und 5.0 unterstützt.

- Leistungseinbußen — Eine Erhöhung der Anzahl von Workern bei der Indexerstellung erhöht die CPU-Auslastung und die Lese-I/O auf der primären Instance Ihrer Amazon DocumentDB DocumentDB-Datenbank. Die Ressourcen, die für die Erstellung eines neuen Indexes benötigt werden, stehen Ihrem laufenden Workload nicht zur Verfügung.
- Elastische Cluster — Parallele Indizierung wird auf elastischen Amazon DocumentDB-Clustern nicht unterstützt.
- Maximale Anzahl an Workern — Die maximale Anzahl von Workern, die Sie konfigurieren können, hängt von der Größe Ihrer primären Instance in Ihrem Datenbank-Cluster ab. Das ist die Hälfte der Gesamtzahl von v CPUs auf der primären Instance Ihres Datenbank-Clusters. Sie können beispielsweise maximal 32 Worker auf einer db.r6g.16xlarge-Instance mit 64 v ausführen. CPUs

  > **Note**
  >
  > Parallele Worker werden in den Instance-Klassen 2xlarge und niedriger nicht unterstützt.

- Mindestanzahl an Workern — Die Mindestanzahl an Workern, die Sie konfigurieren können, ist eins. Die Standardeinstellung für die Indexerstellung auf instanzbasierten Clustern ist zwei Worker. Sie können die Anzahl der Worker jedoch auf einen reduzieren, indem Sie die Option „Worker-Threads“ verwenden. Dadurch wird der Prozess mit einem einzigen Worker ausgeführt.
- Indexkomprimierung — Amazon DocumentDB unterstützt keine Indexkomprimierung. Die Datengrößen für Indizes sind möglicherweise größer als bei Verwendung anderer Optionen.
- Indizierung mehrerer Sammlungen — Die Hälfte des V CPUs auf der primären Instanz Ihres Datenbank-Clusters kann für konfigurierte Worker verwendet werden, die die Indexerstellung für mehrere Sammlungen durchführen.
- Indextypen — In diesem Blogbeitrag finden Sie eine vollständige Erklärung der unterstützten Indextypen in Amazon DocumentDB.

Erste Schritte

Verwenden Sie den `createIndexes` Befehl, um die Indexerstellung für eine Sammlung zu starten. Standardmäßig führt der Befehl zwei parallel Worker aus, wodurch die Geschwindigkeit des Indexerstellungsprozesses um das Zweifache erhöht wird.

Der folgende Befehlsprozess zeigt beispielsweise, wie ein Index für das Feld „user_name“ in einem Dokument erstellt und die Geschwindigkeit des Indizierungsvorgangs auf vier Worker erhöht wird:

1. Erstellen Sie Indizes mit zwei parallel Workern auf dem Cluster:

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
 "name":"username_idx"}]})
```

2. Um die Geschwindigkeit der Indexerstellung zu optimieren, können Sie die Anzahl der Worker mithilfe der Option „Worker-Threads“ (`"workers":<number>`) im `db.runCommand` `createIndexes` Befehl angeben.

   Erhöhen Sie die Geschwindigkeit des Prozesses auf vier parallel Mitarbeiter:

```
db.runCommand({"createIndexes":"test","indexes":[{"key": {"user_name":1},
 "name":"username_idx", "workers":4}]})
```

   > **Note**
   >
   > Je höher die Anzahl der Mitarbeiter, desto schneller schreitet die Indexerstellung voran. Je höher jedoch die Anzahl der Worker steigt, desto höher steigt die Auslastung der V CPUs - und Read-I/O Ihrer primären Instance. Stellen Sie sicher, dass Ihr Cluster ausreichend bereitgestellt ist, um die erhöhte Belastung zu bewältigen, ohne andere Workloads zu beeinträchtigen.

Status des Indizierungsfortschritts

Bei der Indexerstellung werden Sammlungen initialisiert, gescannt, Schlüssel sortiert und schließlich Schlüssel mithilfe eines Indexgenerators eingefügt. Der Prozess besteht aus bis zu sechs Phasen, wenn Sie ihn im Vordergrund ausführen, und bis zu neun Phasen, wenn Sie ihn im Hintergrund ausführen. Sie können Statusmetriken wie die prozentuale Fertigstellung, die Gesamtzahl der gescannten Speicherblöcke, die sortierten Schlüssel und die eingefügten Schlüssel Schritt für Schritt anzeigen.

Überwachen Sie den Fortschritt des Indizierungsprozesses mithilfe des `db.currentOp()` Befehls in der Mongo-Shell. Ein 100-prozentiger Abschluss der letzten Phase zeigt, dass alle Indizes erfolgreich erstellt wurden:

```
db.currentOp({"command.createIndexes": { $exists : true } })
```

> **Note**
>
> Das Anzeigen des Fortschrittsstatus der Indizierung wird nur in Amazon DocumentDB 5.0 unterstützt.

Typen der Indexerstellung

Die vier Arten von Indexerstellungen sind:

- Vordergrund — Der Vordergrund-Index-Build blockiert alle anderen Datenbankoperationen, bis der Index erstellt ist. Der Vordergrundbuild von Amazon DocumentDB besteht aus fünf Phasen.
- Vordergrund (einzigartig) — Index-Builds im Vordergrund für ein einzelnes Dokument (eindeutig) blockieren andere Datenbankoperationen wie reguläre Vordergrund-Builds. Im Gegensatz zum einfachen Vordergrund-Build verwendet der Unique-Build eine zusätzliche Stufe (Sortierung von Schlüsseln 2), um nach doppelten Schlüsseln zu suchen. Der (einzigartige) Build im Vordergrund besteht aus sechs Stufen.
- Hintergrund — Die Indexerstellung im Hintergrund ermöglicht die Ausführung anderer Datenbankoperationen im Vordergrund, während der Index erstellt wird. Der Amazon DocumentDB DocumentDB-Hintergrundaufbau besteht aus acht Phasen.
- Hintergrund (einzigartig) — Index-Builds für ein einzelnes Dokument (eindeutig) im Hintergrund ermöglichen die Ausführung anderer Datenbankoperationen im Vordergrund, während der Index erstellt wird. Im Gegensatz zum grundlegenden Hintergrundbuild verwendet der Unique-Build eine zusätzliche Phase (Sortierung von Schlüsseln 2), um nach doppelten Schlüsseln zu suchen. Der (einzigartige) Background-Build besteht aus neun Stufen.

Phasen der Indexerstellung

Stufe	Vordergrund	Vordergrund (einzigartig)	Hintergrund	Hintergrund (einzigartig)
Initialisieren	1	1	1	1

Stufe	Vordergrund	Vordergrund (einzigartig)	Hintergrund	Hintergrund (einzigartig)
Gebäudeindex: initialisieren	2	2	2	2
Gebäudeindex: Sammlung wird gescannt	3	3	3	3
Gebäudeindex: Schlüssel sortieren 1	4	4	4	4
Gebäudeindex: Sortieren von Schlüsseln 2		5		5
Gebäudeindex: Schlüssel einfügen	5	6	5	6
validieren: Index scannen			6	7
validieren: Tupel sortieren			7	8
validieren: Sammlung scannen			8	9

- initialisieren — CreateIndex bereitet den Index Builder vor. Diese Phase sollte sehr kurz sein.
- Index erstellen: Initialisieren — Der Index Builder bereitet die Erstellung des Indexes vor. Diese Phase sollte sehr kurz sein.
- Index erstellen: Sammlung scannen — Der Index Builder führt einen Sammlungsscan durch, um Indexschlüssel zu sammeln. Die Maßeinheit ist „Blöcke“.

> **Note**
>
> Wenn mehr als ein Worker für den Indexaufbau konfiguriert ist, wird dieser in dieser Phase angezeigt. Die Phase „Erfassung des Scannens“ ist die einzige Phase, in der während der Indexerstellung mehrere Worker verwendet werden. In allen anderen Phasen wird ein einziger Worker angezeigt.

- Index erstellen: Schlüssel sortieren 1 — Der Indexgenerator sortiert die gesammelten Indexschlüssel. Die Maßeinheit ist „Schlüssel“.
- Index erstellen: Schlüssel sortieren 2 — Der Index Builder sortiert die gesammelten Indexschlüssel, die toten Tupeln entsprechen. Diese Phase besteht nur für die Erstellung eines eindeutigen Indexes. Die Maßeinheit ist „Schlüssel“.
- Gebäudeindex: Schlüssel einfügen — Der Index Builder fügt Indexschlüssel in den neuen Index ein. Die Maßeinheit ist „Schlüssel“.
- validieren: Index scannen — CreateIndex durchsucht den Index nach Schlüsseln, die validiert werden müssen. Die Maßeinheit ist „Blöcke“.
- validieren: Tupel sortieren — CreateIndex sortiert die Ausgabe der Indexscanphase.
- validieren: Sammlung scannen — CreateIndex scannt die Sammlung, um die Indexschlüssel zu validieren, die in den beiden vorherigen Phasen gefunden wurden. Die Maßeinheit ist „Blöcke“.

Beispiel für die Ausgabe einer Indexerstellung

Im folgenden Ausgabebeispiel (Indexerstellung im Vordergrund) wird der Status der Indexerstellung angezeigt. Das Feld „msg“ fasst den Baufortschritt zusammen, indem es die Phase und den Prozentsatz der Fertigstellung des Builds angibt. Das Feld „Arbeiter“ gibt die Anzahl der Arbeiter an, die in dieser Phase der Indexerstellung eingesetzt wurden. Das Feld „Fortschritt“ zeigt die tatsächlichen Zahlen, die zur Berechnung des Prozentsatzes der Fertigstellung verwendet wurden.

> **Note**
>
> Die Felder „currentIndexBuildName“, „msg“ und „progress“ werden in Amazon DocumentDB Version 4.0 nicht unterstützt.

```
{
    "inprog" : [{
```

```
    …
        "command": {
            "createIndexes": "test",
            "indexes": [{
                "v": 2,
                "key": {
                    "user_name": 1
                },
                "name": "user_name_1"
            }],
            "lsid": {
                "id": UUID(“094d0fba-8f41-4373-82c3-7c4c7b5ff13b”)
            },
            "$db": "test"
        },
        "currentIndexBuildName": user_name_1,
        "msg": "Index Build: building index number_1, stage 6/6 building index:
 656860/1003520 (keys) 65%",
        "workers": 1,
        "progress": {
            "done": 656861,
            "total": 1003520
        },
    …
    ],

    "ok" : 1
}
```

Pflege Amazon DocumentDB DocumentDB-Indizes

Indexverwaltung mit **`reIndex`**

`reIndex`ist ein Befehl, der verwendet wird, um einen Index neu aufzubauen. Er wird normalerweise verwendet, wenn ein Index beschädigt oder ineffizient geworden ist. Im Laufe der Zeit können Indizes aufgrund vieler Aktualisierungen, Einfügungen oder Löschungen ungenutzten Speicherplatz ansammeln, was zu Leistungseinbußen führt. Eine Neuindizierung hilft dabei, diesen ungenutzten Speicherplatz zu entfernen und die Effizienz des Index wiederherzustellen.

`reIndex`Richtlinien

- `reIndex`wird nur auf Amazon DocumentDB 5.0 unterstützt.

- Amazon DocumentDB unterstützt `reindex` einen einzelnen Index im Hintergrund, sodass mehrere Worker verwendet werden können. Der alte Index kann für Abfragen verwendet werden, wenn der `reIndex` Prozess läuft.
- Amazon DocumentDB unterstützt die Indexierung von Fortschrittsberichten bis. `currentOp` Sie können die Phasen der Indexerstellung sehen, die denen bei der Phasen der Indexerstellung Indexerstellung ähneln. Der einzige Unterschied besteht darin, dass es `reIndex` immer acht Stufen gibt, unabhängig davon, ob sie einzigartig sind oder nicht. Es gibt keine Phase „Index erstellen: Schlüssel sortieren 2“.
- `reIndex`kann gleichzeitig mit jedem Befehl ausgeführt werden, mit Ausnahme von indexbezogenen Befehlen in derselben Sammlung:`createIndexes`,`dropIndexes`, `collMod` und. `renameCollection`
- `reIndex`wird derzeit nicht für Text-, Geodaten-, Vektor- und Teilindizes unterstützt.

`reIndex`bauen

Verwenden Sie den folgenden Befehl, um Ihren Index neu zu erstellen:

```
db.runCommand({ reIndex: "collection-name", index: "index-name"})
```

Optional können Sie auch die Anzahl der Mitarbeiter steuern, die dem Neuaufbauprozess zugewiesen sind:

```
db.runCommand({ reIndex: "collection-name", index: "index-name", workers: number })
```

Verwaltung der Dokumentenkomprimierung auf Sammlungsebene

Mit der Dokumentenkomprimierung auf Sammlungsebene von Amazon DocumentDB können Sie die Speicher- und I/O-Kosten senken, indem Sie die Dokumente in Ihren Sammlungen komprimieren. Sie können die Dokumentenkomprimierung auf Sammlungsebene aktivieren und bei Bedarf Komprimierungsmetriken anzeigen, indem Sie den Speicherzuwachs anhand von Komprimierungsmetriken wie der Speichergröße komprimierter Dokumente und dem Komprimierungsstatus messen. Amazon DocumentDB verwendet den LZ4 Komprimierungsalgorithmus, um Dokumente zu komprimieren.

Amazon DocumentDB unterstützt die Dokumentenkomprimierung ab Version 5.0. Im Folgenden finden Sie Funktionen zur Komprimierung von Dokumenten auf Sammlungsebene:

- Standardverhalten — Die Standardkomprimierungseinstellung für neue Sammlungen in einem Cluster wird durch den Cluster-Parameter bestimmt. `default_collection_compression` Dieser Parameter ist standardmäßig auf „deaktiviert“ gesetzt.
- Komprimieren vorhandener Sammlungen — Die Komprimierungseinstellungen für bestehende Sammlungen können mit dem `collMod` Befehl geändert werden.
- Ändern des Komprimierungsschwellenwerts — Der Standardschwellenwert für die Komprimierung beträgt 2 KB. Dieser Wert kann für neue Sammlungen mit dem `createCollection` Befehl angegeben und für bestehende Sammlungen mit dem `collMod` Befehl geändert werden.

> **Note**
>
> Die Amazon DocumentDB DocumentDB-Dokumentenkomprimierung wird in den Amazon DocumentDB DocumentDB-Versionen 3.6 und 4.0 nicht unterstützt.

Themen

- Verwaltung der Dokumentenkomprimierung
- Überwachung der Dokumentenkomprimierung

Verwaltung der Dokumentenkomprimierung

Aktivieren der Dokumentenkomprimierung in einer Sammlung

Aktivieren Sie die Dokumentenkomprimierung beim Erstellen einer Sammlung auf Amazon DocumentDB mithilfe `db.createCollection()` der folgenden Methode:

```
db.createCollection( sample_collection,{
    storageEngine : {
        documentDB: {
            compression:{enable: <true | false>}
        }
    }
})
```

Aktivieren der Dokumentenkomprimierung in einem Cluster

Die Dokumentenkomprimierung kann standardmäßig für alle neuen Sammlungen auf Clusterebene aktiviert werden, indem der Cluster-Parameter `default_collection_compression` auf „enabled“ gesetzt wird. Wenn dieser Parameter auf „aktiviert“ gesetzt ist, ist für neu erstellte Sammlungen auf dem Cluster die Komprimierung standardmäßig aktiviert, wobei der Komprimierungsschwellenwert 2 KB beträgt.

Komprimieren vorhandener Sammlungen

Sie können auch die Komprimierungseinstellungen für eine vorhandene Sammlung ändern, indem Sie den `collMod` Vorgang verwenden und die folgende `storageEngine` Konfiguration angeben. Bitte beachten Sie, dass die mit diesem Befehl vorgenommene Änderung nur für neu eingefügte/aktualisierte Dokumente gilt und dass sich die Komprimierung zuvor eingefügter Dokumente nicht ändert.

```
db.runCommand({
    collMod: "orders",
    storageEngine: {
        documentDB: {compression: {enable: <true | false>} }
    }
})
```

Einstellung der Schwellenwerte für die Komprimierung

Standardmäßig liegt der Komprimierungsschwellenwert für komprimierte Sammlungen bei 2032 Byte. Dieser Schwellenwert kann im `createCollection` Befehl festgelegt werden, wenn eine neue Sammlung mit aktivierter Komprimierung erstellt wird:

```
db.createCollection( sample_collection, {
    storageEngine : {
        documentDB: {
            compression: {
                enable: true,
                threshold: <128 - 8000>
            }
        }
    }
})
```

Sie können den Komprimierungsschwellenwert für eine vorhandene komprimierte Sammlung auch ändern, indem Sie den `collMod` Vorgang verwenden und die folgende `storageEngine` Konfiguration angeben:

```
db.runCommand({
    collMod: "orders",
    storageEngine: {
        documentDB: {
            compression: {
                enable: true,
                threshold: <128 - 8000>
            }
        }
    }
})
```

Bitte beachten Sie, dass der Komprimierungsschwellenwert nur auf einen Wert zwischen 128 und 8000 Byte festgelegt werden kann. Außerdem muss die `enable` Option bei der Angabe des Komprimierungsschwellenwerts auf „true“ gesetzt werden.

Überwachung der Dokumentenkomprimierung

Sie können überprüfen, ob eine Sammlung komprimiert ist, und ihr Komprimierungsverhältnis wie folgt berechnen.

Zeigen Sie die Komprimierungsstatistiken an, indem Sie den `db.collection.stats()` Befehl `db.printCollectionStats()` oder in der Mongo-Shell ausführen. Die Ausgabe zeigt Ihnen die Originalgröße und die komprimierte Größe, die Sie vergleichen können, um die Speichergewinne durch die Dokumentenkomprimierung zu analysieren. In diesem Beispiel werden die Statistiken für eine Sammlung mit dem Namen „sample_collection“ unten angezeigt. Im Folgenden wird ein Skalierungsfaktor von 1024*1024 verwendet, um die Werte und in MB auszugeben. `size` `storageSize`

```
db.sample_collection.stats(1024*1024)
```

Das Folgende ist ein Beispiel für die Ausgabe für den obigen Befehl:

```
{
    "ns" : "test.sample_collection",
```

```
    "count" : 1000000,
    "size" : 3906.3,
    "avgObjSize" : 4096,
    "storageSize" : 1953.1,
    compression:{"enabled" : true,"threshold" : 2032},
    ...
}
```

- Größe — Die Originalgröße der Dokumentensammlung.
- avgObjSize- Die durchschnittliche Dokumentgröße vor der Komprimierung, abgerundet auf die erste Dezimalzahl. Die Maßeinheit ist Byte.
- StorageSize — Die Speichergröße der Sammlung nach der Komprimierung. Die Maßeinheit ist Byte.
- aktiviert — Zeigt an, ob die Komprimierung aktiviert oder deaktiviert ist.

Um das tatsächliche Komprimierungsverhältnis zu berechnen, dividieren Sie die Sammlungsgröße durch die Speichergröße (Size/StorageSize). Für das obige Beispiel lautet die Berechnung 3906,3/1953,1, was einem Komprimierungsverhältnis von 2:1 entspricht.

Amazon DocumentDB DocumentDB-Ereignisse verwalten

Amazon DocumentDB (mit MongoDB-Kompatibilität) zeichnet Ereignisse auf, die sich auf Ihre Cluster, Instances, Snapshots, Sicherheitsgruppen und Cluster-Parametergruppen beziehen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp des Ereignisses und eine dem Ereignis zugehörige Benachrichtigung.

> **Important**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS und Amazon Neptune gemeinsam genutzt wird. Regionsgrenzen, Grenzwerte, die auf Regionsebene geregelt werden, werden von Amazon DocumentDB, Amazon RDS und Amazon Neptune gemeinsam genutzt. Weitere Informationen finden Sie unter Regionale Kontingente.

Themen

- Amazon DocumentDB DocumentDB-Ereigniskategorien anzeigen

- Amazon DocumentDB DocumentDB-Ereignisse anzeigen

Amazon DocumentDB DocumentDB-Ereigniskategorien anzeigen

Jeder Amazon DocumentDB DocumentDB-Ressourcentyp hat bestimmte Ereignistypen, die ihm zugeordnet werden können. Sie können den AWS CLI `describe-event-categories` Vorgang verwenden, um die Zuordnung zwischen Ereignistypen und Amazon DocumentDB DocumentDB-Ressourcentypen anzuzeigen.

Parameter

- **`--source-type`**— Fakultativ. Verwenden Sie den `--source-type`-Parameter, um die Ereigniskategorien für einen bestimmten Quelltyp anzuzeigen. Die folgenden Werte sind zulässig:
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`
- **`--filters`**—Fakultativ. Verwenden Sie den Filter`--filter Name=engine,Values=docdb`, um die Ereigniskategorien nur für Amazon DocumentDB anzuzeigen.

Example

Der folgende Code listet die Ereigniskategorien im Zusammenhang mit Clustern auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-event-categories \
    --filter Name=engine,Values=docdb \
    --source-type db-cluster
```

Für Windows:

```
aws docdb describe-event-categories ^
    --filter Name=engine,Values=docdb ^
    --source-type db-cluster
```

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

```
{
    "EventCategoriesMapList": [
        {
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
            ],
            "SourceType": "db-cluster"
        }
    ]
}
```

Der folgende Code listet die Ereigniskategorien auf, die jedem Amazon DocumentDB DocumentDB-Quelltyp zugeordnet sind.

```
aws docdb describe-event-categories
```

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

```
{
    "EventCategoriesMapList": [
        {
            "SourceType": "db-instance",
            "EventCategories": [
                "notification",
                "failure",
                "creation",
                "maintenance",
                "deletion",
                "recovery",
                "restoration",
                "configuration change",
                "read replica",
                "backtrack",
                "low storage",
                "backup",
                "availability",
                "failover"
```

```
            ]
        },
        {
            "SourceType": "db-security-group",
            "EventCategories": [
                "configuration change",
                "failure"
            ]
        },
        {
            "SourceType": "db-parameter-group",
            "EventCategories": [
                "configuration change"
            ]
        },
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "notification",
                "failure",
                "maintenance",
                "failover"
            ]
        },
        {
            "SourceType": "db-cluster-snapshot",
            "EventCategories": [
                "backup"
            ]
        }
    ]
}
```

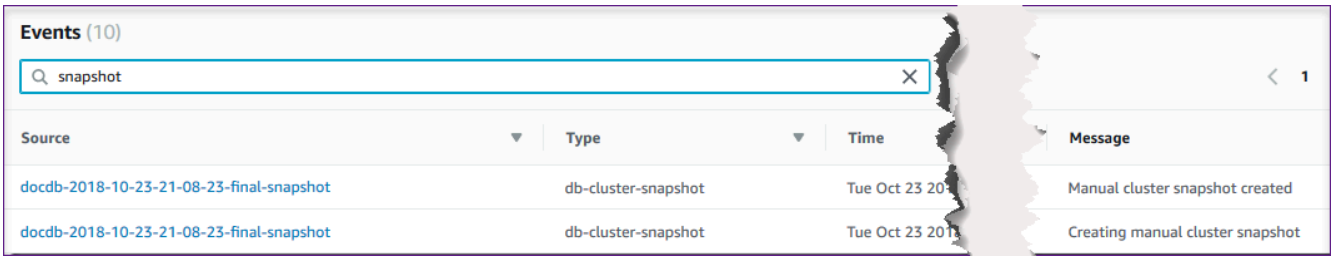
Amazon DocumentDB DocumentDB-Ereignisse anzeigen

Sie können Ereignisse für Ihre Amazon DocumentDB DocumentDB-Ressourcen über die Amazon DocumentDB DocumentDB-Konsole abrufen, in der Ereignisse der letzten 24 Stunden angezeigt werden. Sie können auch Ereignisse für Ihre Amazon DocumentDB DocumentDB-Ressourcen abrufen, indem Sie den AWS CLI Befehl describe-events oder den DescribeEventsAmazon DocumentDB DocumentDB-API-Vorgang verwenden. Wenn Sie die AWS CLI oder die Amazon DocumentDB DocumentDB-API verwenden, um Ereignisse anzuzeigen, können Sie Ereignisse der letzten 14 Tage abrufen.

Using the AWS Management Console

Um alle Amazon DocumentDB DocumentDB-Instance-Ereignisse der letzten 24 Stunden anzuzeigen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich die Option Events. Die verfügbaren Ereignisse erscheinen in einer Liste.
3. Verwenden Sie die Liste Filter, um die Ereignisse nach Typ zu filtern. Geben Sie einen Begriff in das Textfeld ein, um Ihre Ergebnisse weiter zu filtern. Der folgende Screenshot zeigt beispielsweise, wie alle Amazon DocumentDB DocumentDB-Ereignisse nach Snapshot-Ereignissen gefiltert werden.



Using the AWS CLI

Um alle Amazon DocumentDB DocumentDB-Instance-Ereignisse der letzten 7 Tage anzuzeigen

Sie können alle Amazon DocumentDB DocumentDB-Instance-Ereignisse der letzten 7 Tage anzeigen, indem Sie den AWS CLI Vorgang describe-events ausführen, wobei der `--duration` Parameter auf `10080` (10.080 Minuten) gesetzt ist.

```
aws docdb describe-events --duration 10080
```

Filtern nach Amazon DocumentDB DocumentDB-Ereignissen

Um bestimmte Amazon DocumentDB DocumentDB-Ereignisse zu sehen, verwenden Sie den `describe-events` Vorgang mit den folgenden Parametern.

Parameter

- **`--filter`**— Erforderlich, um zurückgegebene Werte auf Amazon DocumentDB DocumentDB-Ereignisse zu beschränken. Wird verwendet**`Name=engine,Values=docdb`**, um alle Ereignisse nur für Amazon DocumentDB zu filtern.
- **`--source-identifier`**— Fakultativ. ID der Ereignisquelle, für die Ereignisse zurückgegeben werden. Wenn diese Option weggelassen wird, werden Ereignisse aus allen Quellen in die Ergebnisse einbezogen.
- **`--source-type`**— Optional, sofern nicht `--source-identifier` angegeben, dann erforderlich. Wenn `--source-identifier` angegeben wird, muss `--source-type` mit dem Typ der `--source-identifier` übereinstimmen. Die folgenden Werte sind zulässig:
  - `db-cluster`
  - `db-instance`
  - `db-parameter-group`
  - `db-security-group`
  - `db-cluster-snapshot`

Das folgende Beispiel listet alle Ihre Amazon DocumentDB DocumentDB-Ereignisse auf.

```
aws docdb describe-events --filters Name=engine,Values=docdb
```

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

```
{
    "Events": [
        {
            "SourceArn": "arn:aws:rds:us-east-1:123SAMPLE012:db:sample-cluster-
instance3",
            "Message": "instance created",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:17:40.023Z",
            "SourceIdentifier": "sample-cluster-instance3",
            "EventCategories": [
                "creation"
            ]
        },
        {
```

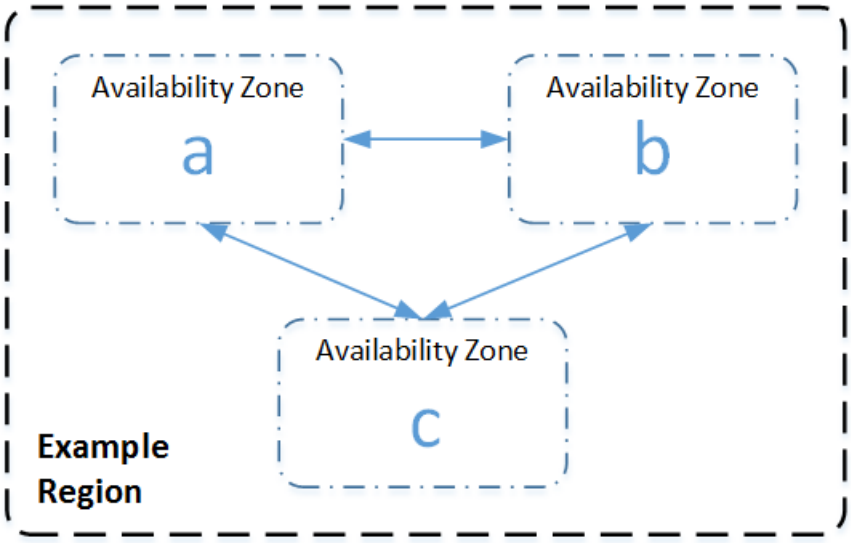
```
            "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
            "Message": "instance shutdown",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:25:01.245Z",
            "SourceIdentifier": "docdb-2018-12-11-21-08-23",
            "EventCategories": [
                "availability"
            ]
        },
        {
            "SourceArn": "arn:aws:rds:us-
east-1:123SAMPLE012:db:docdb-2018-12-11-21-08-23",
            "Message": "instance restarted",
            "SourceType": "db-instance",
            "Date": "2018-12-11T21:25:11.441Z",
            "SourceIdentifier": "docdb-2018-12-11-21-08-23",
            "EventCategories": [
                "availability"
            ]
        }
    ]
}
```

Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen.

Auswahl von Regionen und Availability Zones

Amazon Cloud Computing-Ressourcen werden an mehreren Standorten weltweit gehostet. Diese Standorte bestehen aus AWS-Regionen Availability Zones. Jeder AWS-Regionist ein separates geografisches Gebiet. Jede Region verfügt über mehrere isolierte Standorte, die als Availability Zones bezeichnet werden. Amazon DocumentDB bietet Ihnen die Möglichkeit, Ressourcen wie Instances und Daten an mehreren Standorten zu platzieren. Ressourcen werden nur repliziert, AWS-Regionen wenn Sie dies ausdrücklich tun.

Amazon betreibt hochmoderne, hoch verfügbare Rechenzentren. In seltenen Fällen kann es aber zu Ausfällen kommen, die die Verfügbarkeit von Instances desselben Standorts beeinträchtigen. Wenn Sie Ihre gesamten Instances an einem einzelnen Standort hosten, der von einem Ausfall dieser Art betroffen ist, ist keine Ihrer Instances verfügbar. Das folgende Diagramm zeigt eine AWS-Region mit drei Availability Zones.



Beachten Sie, dass alle Regionen voneinander unabhängig sind. Jede Amazon DocumentDB DocumentDB-Aktivität, die Sie initiieren (z. B. Instances erstellen oder verfügbare Instances auflisten), wird nur in Ihrer aktuellen Standardeinstellung AWS-Region ausgeführt. Sie können die Standardregion in der Konsole ändern, indem Sie die Umgebungsvariable `EC2_REGION` festlegen. Oder Sie können sie überschreiben, indem Sie den Parameter `--region` in der AWS CLI verwenden. Weitere Informationen finden Sie unter Konfiguration der AWS Command Line Interface Abschnitte zu Umgebungsvariablen und Befehlszeilenoptionen.

Wenn Sie mit der Amazon DocumentDB-Konsole einen Cluster erstellen und sich dafür entscheiden, ein Replikat in einer anderen Availability Zone zu erstellen, erstellt Amazon DocumentDB zwei Instances. Es erstellt die primäre Instance in einer Availability Zone und die Replikat-Instance in einer anderen Availability Zone. Das Cluster-Volumen wird immer über drei Availability Zones repliziert.

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

Verfügbarkeit in Regionen

Amazon DocumentDB ist in den folgenden AWS Regionen verfügbar.

Von Amazon DocumentDB unterstützte Regionen

Name der Region	Region	Verfügbarkeitszonen (Datenverarbeitung)
USA Ost (Ohio)	`us-east-2`	3

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

Name der Region	Region	Verfügbarkeitszonen (Datenverarbeitung)
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
Europa (Spain)	`eu-south-2`	3
Europa (Stockholm)	`eu-north-1`	3
Naher Osten (VAE)	`me-central-1`	3
Israel (Tel Aviv)	`il-central-1`	3
AWS GovCloud (USA West)	`us-gov-west-1`	3
AWS GovCloud (US-Ost)	`us-gov-east-1`	3

Standardmäßig ist die Zeitzone für einen Amazon DocumentDB-Cluster Universal Time Coordinated (UTC).

Informationen zum Suchen der Verbindungsendpunkte für Cluster und Instances in einer bestimmten Region finden Sie unter Grundlegendes zu Amazon DocumentDB DocumentDB-Endpunkten.

Verwaltung von Amazon DocumentDB-Cluster-Parametergruppen

Sie können die Amazon DocumentDB DocumentDB-Engine-Konfiguration verwalten, indem Sie Parameter in einer Cluster-Parametergruppe verwenden. Eine Cluster-Parametergruppe ist eine Sammlung von Amazon DocumentDB-Konfigurationswerten, die die Verwaltung der Parameter Ihrer Amazon DocumentDB-Cluster erleichtern. Cluster-Parametergruppen dienen als Container für Engine-Konfigurationswerte, die auf jede alle Instances im Cluster angewendet werden.

In diesem Abschnitt wird beschrieben, wie Cluster-Parametergruppen erstellt, angezeigt und geändert werden. Es wird auch gezeigt, wie Sie ermitteln können, welche Cluster-Parametergruppe einem bestimmten Cluster zugeordnet ist.

Themen

- Beschreibung von Amazon DocumentDB-Cluster-Parametergruppen
- Amazon DocumentDB-Cluster-Parametergruppen erstellen
- Amazon DocumentDB-Cluster-Parametergruppen ändern
- Ändern von Amazon DocumentDB-Clustern zur Verwendung benutzerdefinierter Cluster-Parametergruppen
- Amazon DocumentDB-Cluster-Parametergruppen kopieren
- Amazon DocumentDB-Cluster-Parametergruppen zurücksetzen
- Löschen von Amazon DocumentDB-Cluster-Parametergruppen
- Referenz zu den Amazon DocumentDB-Clusterparametern

Beschreibung von Amazon DocumentDB-Cluster-Parametergruppen

Eine `default` Cluster-Parametergruppe wird automatisch erstellt, wenn Sie den ersten Amazon DocumentDB-Cluster in einer neuen Region erstellen oder eine neue Engine verwenden. Nachfolgende Cluster, die in derselben Region erstellt werden und dieselbe Engine-Version haben, werden mit der `default` Cluster-Parametergruppe erstellt.

Themen

- Beschreibung der Details einer Amazon DocumentDB-Cluster-Parametergruppe
- Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters

Beschreibung der Details einer Amazon DocumentDB-Cluster-Parametergruppe

Um die Details einer bestimmten Cluster-Parametergruppe zu beschreiben, führen Sie die folgenden Schritte mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) aus.

Using the AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) den Namen der Parametergruppe aus, deren Details Sie anzeigen möchten.
4. Auf der anschließend angezeigten Seite werden die Parameter der Parametergruppe, die letzte Aktivität und Tags angezeigt.

   - In Cluster parameters (Cluster-Parameter) werden Ihnen der Name, der aktuelle Wert, die zulässigen Werte, der Anwendungstyp, der Datentyp und die Beschreibung des Parameters angezeigt. Sie können auch erkennen, ob der Parameter geändert werden kann. Sie können einzelne Parameter ändern, indem Sie den Parameter auswählen und dann im Abschnitt Cluster-Parameter auf Bearbeiten klicken. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parameter ändern.
   - In Recent events (Aktuelle Ereignisse) werden Ihnen die jeweils aktuellen Ereignisse für diese Parametergruppe angezeigt. Sie können diese Ereignisse über die Suchleiste in diesem Abschnitt filtern. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse verwalten.
   - Unter Tags werden die Tags angezeigt, die sich in dieser Cluster-Parametergruppe befinden. Sie können Tags hinzufügen oder entfernen, indem Sie im Abschnitt „Tags“ die Option „Bearbeiten“ wählen. Weitere Informationen finden Sie unter Taggen von Amazon DocumentDB DocumentDB-Ressourcen.

Using the AWS CLI

Sie können den `describe-db-cluster-parameter-groups` AWS CLI Befehl verwenden, um den Amazon-Ressourcennamen (ARN), die Familie, die Beschreibung und den Namen einer einzelnen Cluster-Parametergruppe oder aller Cluster-Parametergruppen, die Sie für Amazon DocumentDB haben, anzuzeigen. Sie können den `describe-db-cluster-parameters` AWS CLI Befehl auch verwenden, um die Parameter und ihre Details innerhalb einer einzelnen Cluster-Parametergruppe anzuzeigen.

- **`--describe-db-cluster-parameter-groups`**— Um eine Liste all Ihrer Cluster-Parametergruppen und ihrer Details zu sehen.
  - **`--db-cluster-parameter-group-name`**— Fakultativ. Der Name der Cluster-Parametergruppe, die beschrieben werden soll. Wenn dieser Parameter weggelassen wird, werden alle Cluster-Parametergruppen beschrieben.
- **`--describe-db-cluster-parameters`**— Um alle Parameter innerhalb einer Parametergruppe und ihre Werte aufzulisten.
  - **`--db-cluster-parameter-group name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die beschrieben werden soll.

Example

Der folgende Code listet bis zu 100 Cluster-Parametergruppen zusammen mit ARN, Familie, Beschreibung und Name auf.

```
aws docdb describe-db-cluster-parameter-groups
```

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

```
{
          "DBClusterParameterGroups": [
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:012345678912:cluster-pg:default.docdb4.0",
                  "DBParameterGroupFamily": "docdb4.0",
                  "Description": "Default cluster parameter group for docdb4.0",
                  "DBClusterParameterGroupName": "default.docdb4.0"
              },
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:012345678912:cluster-pg:sample-parameter-group",
                  "DBParameterGroupFamily": "docdb4.0",
                  "Description": "Custom docdb4.0 parameter group",
                  "DBClusterParameterGroupName": "sample-parameter-group"
              }
          ]
}
```

Example

Der folgende Code listet den ARN, die Familie, die Beschreibung und den Namen für `sample-parameter-group` auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
          --db-cluster-parameter-group-name sample-parameter-group
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
          --db-cluster-parameter-group-name sample-parameter-group
```

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

```
{
          "DBClusterParameterGroups": [
              {
                  "DBClusterParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:cluster-pg:sample-parameter-group",
                  "Description": "Custom docdb4.0 parameter group",
                  "DBParameterGroupFamily": "docdb4.0",
                  "DBClusterParameterGroupName": "sample-parameter-group"
              }
          ]
}
```

Example

Der folgende Code listet die Werte der Parameter in `sample-parameter-group` auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

Für Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

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

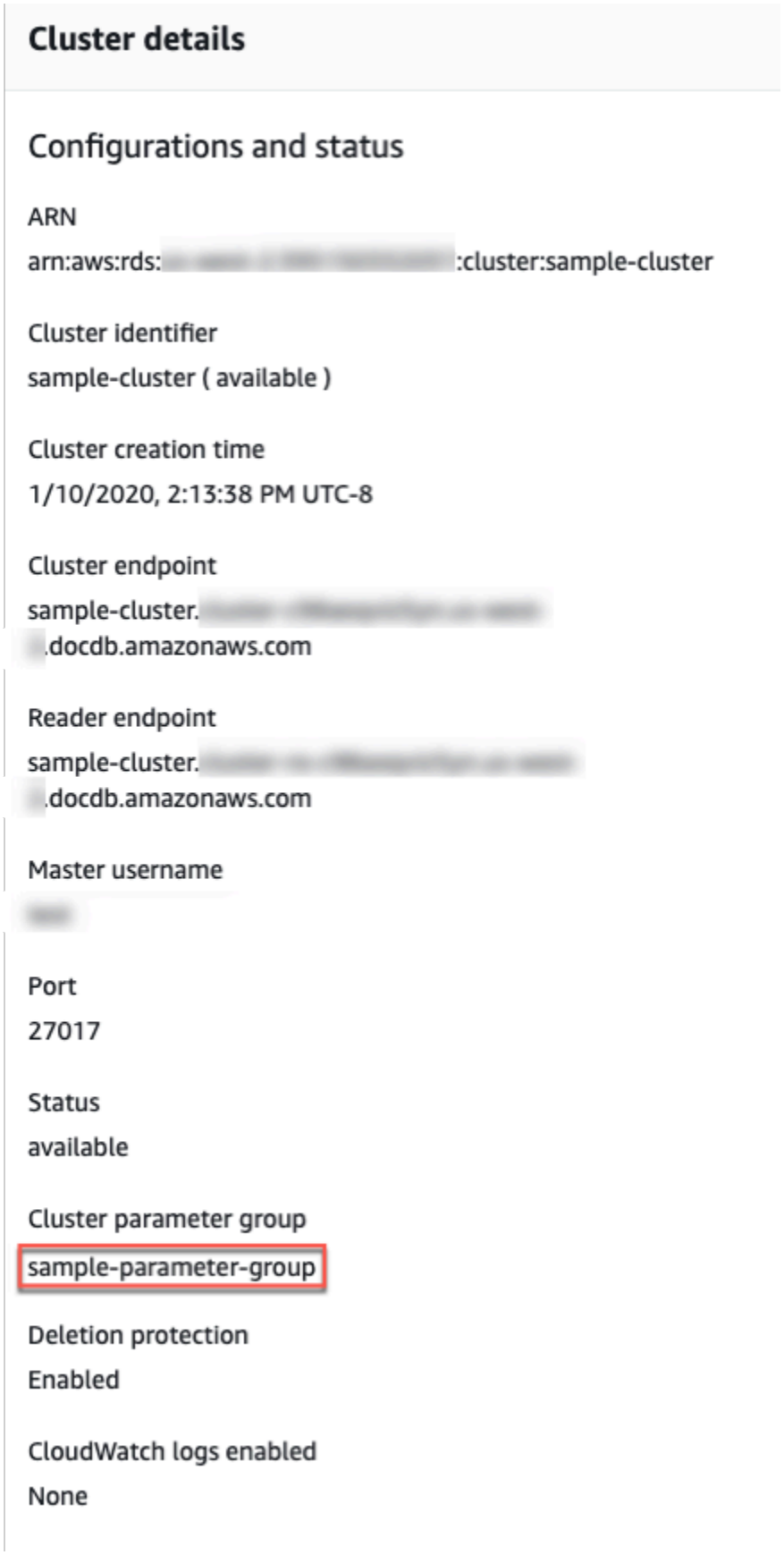
```
{
   "Parameters": [
         {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
         },
         {
            "ParameterName": "change_stream_log_retention_duration",
            "ParameterValue": "17777",
            "Description": "Duration of time in seconds that the change stream log
 is retained and can be consumed.",
            "Source": "user",
            "ApplyType": "dynamic",
            "DataType": "integer",
            "AllowedValues": "3600-86400",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
         }
   ]
}
```

Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters

Um zu ermitteln, welche Parametergruppe einem bestimmten Cluster zugeordnet ist, führen Sie die folgenden Schritte mit dem AWS Management Console oder dem aus AWS CLI.

Using the AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im linken Navigationsbereich Cluster aus.
3. Wählen Sie aus der Liste der Cluster den Namen des Clusters aus, an dem Sie interessiert sind.
4. Auf der anschließend angezeigten Seite werden die Details des von Ihnen ausgewählten Clusters angezeigt. Scrollen Sie nach unten zu Cluster details (Clusterdetails). Sie finden den Namen der Parametergruppe unten in diesem Abschnitt unterhalb von Cluster parameter group (Cluster-Parametergruppe).


Cluster details
Configurations and status
ARN
arn:aws:rds:
:cluster:sample-cluster
Cluster identifier
sample-cluster (available)
Cluster creation time
1/10/2020, 2:13:38 PM UTC-8
Cluster endpoint
sample-cluster.
.docdb.amazonaws.com
Reader endpoint
sample-cluster.
.docdb.amazonaws.com
Master username
Port
27017
Status
available
Cluster parameter group
sample-parameter-group
Deletion protection
Enabled
CloudWatch logs enabled
None

Using the AWS CLI

Der folgende AWS CLI Code bestimmt, welche Parametergruppe den Cluster steuert. `sample-cluster`

```
aws docdb describe-db-clusters \
      --db-cluster-identifier sample-cluster \
      --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

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

```
[
       [
           "sample-cluster",
           "sample-parameter-group"
       ]
]
```

Amazon DocumentDB-Cluster-Parametergruppen erstellen

Standard-Cluster-Parametergruppen wie`default.docdb5.0`, oder`default.docdb4.0`, werden erstellt`default.docdb3.6`, wenn Sie einen Cluster mit einer neuen Engine-Version und in einer neuen Region erstellen. Nachfolgende Cluster, die in dieser Region und mit derselben Engine-Version erstellt wurden, erben die `default` Cluster-Parametergruppe. Nach der Erstellung können die `default` Parametergruppen nicht gelöscht oder umbenannt werden. Sie können das Engine-Verhalten von Cluster-Instances ändern, indem Sie eine benutzerdefinierte Parametergruppe mit bevorzugten Parameterwerten erstellen und sie an Ihren Amazon DocumentDB-Cluster anhängen.

Das folgende Verfahren führt Sie durch die Erstellung einer benutzerdefinierten Cluster-Parametergruppe. Anschließend können Sie die Parameter innerhalb dieser Parametergruppe ändern.

> **Note**
>
> Sie sollten nach der Erstellung einer Cluster-Parametergruppe mindestens 5 Minuten warten, bevor Sie diese Cluster-Parametergruppe verwenden. Dadurch kann Amazon DocumentDB die `create` Aktion vollständig abschließen, bevor die Cluster-Parametergruppe für einen neuen Cluster verwendet wird. Sie können die Operation AWS Management Console oder die `describe-db-cluster-parameter-groups` AWS CLI Operation verwenden, um zu

überprüfen, ob Ihre Cluster-Parametergruppe erstellt wurde. Weitere Informationen finden Sie unter Beschreibung von Amazon DocumentDB-Cluster-Parametergruppen.

Using the AWS Management Console

So erstellen Sie eine Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) die Option Create (Erstellen) aus.
4. Geben Sie im Bereich Create cluster parameter group (Cluster-Parametergruppe erstellen) Folgendes ein:

   a. Gruppenname — Geben Sie einen Namen für die Cluster-Parametergruppe ein. Beispiel, `sample-parameter-group`. Für Cluster-Parametergruppen gelten die folgenden Benennungseinschränkungen:

      - Die Länge muss [1 bis 255] alphanumerische Zeichen betragen.
      - Muss mit einem Buchstaben beginnen.
      - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

   b. Beschreibung — Geben Sie eine Beschreibung für diese Cluster-Parametergruppe ein.
5. Um die Cluster-Parametergruppe zu erstellen, wählen Sie Erstellen aus. Um die Operation abzubrechen, wählen Sie Abbrechen aus.
6. Nach der Auswahl von Create (Erstellen) wird oben auf der Seite der folgende Text angezeigt, um zu bestätigen, dass die Cluster-Parametergruppe erfolgreich erstellt wurde:

```
Successfully created cluster parameter group 'sample-parameter-group'.
```

Using the AWS CLI

Um eine neue Cluster-Parametergruppe für Amazon DocumentDB 4.0-Cluster zu erstellen, verwenden Sie den AWS CLI `create-db-cluster-parameter-group` Vorgang mit den folgenden Parametern:

- **`--db-cluster-parameter-group-name`**— Der Name der benutzerdefinierten Cluster-Parametergruppe. Beispiel, `sample-parameter-group`.
- **`--db-cluster-parameter-group-family`**— Die Cluster-Parametergruppen-Familie, die als Vorlage für die benutzerdefinierte Cluster-Parametergruppe verwendet wird. Derzeit muss diese `docdb4.0` lauten.
- **`--description`**— Die vom Benutzer bereitgestellte Beschreibung für diese Cluster-Parametergruppe. Das folgende Beispiel verwendet "`Custom docdb4.0 parameter group`“.

Für Linux, macOS oder Unix:

Example

```
aws docdb create-db-cluster-parameter-group \
 --db-cluster-parameter-group-name sample-parameter-group \
 --db-parameter-group-family docdb4.0 \
 --description "Custom docdb4.0 parameter group"
```

Für Windows:

```
aws docdb create-db-cluster-parameter-group ^
 --db-cluster-parameter-group-name sample-parameter-group ^
 --db-parameter-group-family docdb4.0 ^
 --description "Custom docdb4.0 parameter group"
```

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

```
{
```

```
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "sample-parameter-group",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Custom docdb4.0 parameter group",
        "DBClusterParameterGroupArn": "sample-parameter-group-arn"
    }
}
```

Amazon DocumentDB-Cluster-Parametergruppen ändern

In diesem Abschnitt wird erklärt, wie Sie eine benutzerdefinierte Amazon DocumentDB DocumentDB-Parametergruppe ändern. In Amazon DocumentDB können Sie eine `default` Cluster-Parametergruppe nicht ändern, die erstellt wird, wenn Sie zum ersten Mal einen Cluster mit einer neuen Engine-Version in einer neuen Region erstellen. Wenn Ihr Amazon DocumentDB-Cluster die Standard-Cluster-Parametergruppe verwendet und Sie einen Wert darin ändern möchten, müssen Sie zuerst eine neue Parametergruppe erstellen oder eine bestehende Parametergruppe kopieren, sie ändern und dann die geänderte Parametergruppe auf Ihren Cluster anwenden.

Gehen Sie wie folgt vor, um eine benutzerdefinierte Cluster-Parametergruppe zu ändern. Die Übertragung von Änderungsaktionen kann eine Weile dauern. Bitte warten Sie, bis die geänderte Cluster-Parametergruppe verfügbar ist, bevor Sie sie an Ihren Cluster anhängen. Sie können den AWS CLI `describe-db-cluster-parameters` Vorgang AWS Management Console oder verwenden, um zu überprüfen, ob Ihre Cluster-Parametergruppe geändert wurde. Weitere Informationen finden Sie unter Beschreibung von Cluster-Parametergruppen.

Using the AWS Management Console

Gehen Sie wie folgt vor, um eine benutzerdefinierte Amazon DocumentDB DocumentDB-Parametergruppe zu ändern. Sie können eine `default`-Parametergruppe nicht ändern. Wenn Sie einen Wert in der `default`-Parametergruppe ändern möchten, können Sie die standardmäßige Cluster-Parametergruppe kopieren, sie ändern und die geänderte Parametergruppe anschließend auf Ihren Cluster anwenden. Weitere Hinweise zum Anwenden von Parametergruppen auf Ihren Cluster finden Sie unter Ändern eines Amazon DocumentDB-Clusters.

So ändern Sie eine benutzerdefinierte Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Parameter groups (Parametergruppen) aus. Wählen Sie in der Liste der Parametergruppen den Namen der Parametergruppe, die Sie ändern möchten.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Gehen Sie bei jedem Parameter in der Parametergruppe, die Sie ändern möchten, wie folgt vor:

   a. Suchen Sie den Parameter, den Sie ändern möchten, und überprüfen Sie, ob er änderbar ist, indem Sie überprüfen, ob er in der **`true`** Spalte Modifizierbar als aufgeführt ist.

   b. Wenn der Parameter geändert werden kann, wählen Sie ihn aus und wählen oben rechts auf der Konsolenseite Edit (Bearbeiten) aus.

   c. Nehmen Sie im Dialogfeld Modify **`<parameter-name>`** die gewünschten Änderungen vor. Wählen Sie anschließend Modify cluster parameter (Cluster-Parameter ändern) aus. Um die Änderungen zu verwerfen, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Sie können den `ParameterValueDescription`, oder `ApplyMethod` eines beliebigen modifizierbaren Parameters in einer benutzerdefinierten Amazon DocumentDB-Cluster-Parametergruppe ändern, indem Sie den. AWS CLI Sie können eine standardmäßige Cluster-Parametergruppe nicht direkt ändern.

Um die Parameter einer benutzerdefinierten Cluster-Parametergruppe zu ändern, verwenden Sie die Operation `modify-db-cluster-parameter-group` mit den folgenden Parametern.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die Sie ändern.
- **`--parameters`** – Erforderlich. Die Parameter, die von Ihnen geändert werden. Eine Liste der Parameter, die für alle Instances in einem Amazon DocumentDB-Cluster gelten, finden Sie

unter. Referenz zu den Amazon DocumentDB-Clusterparametern Jeder Parametereintrag muss Folgendes enthalten:

- **`ParameterName`**— Der Name des Parameters, den Sie ändern.
- **`ParameterValue`**— Der neue Wert für diesen Parameter.
- **`ApplyMethod`**— Wie die Änderungen an diesem Parameter angewendet werden sollen. Zugelassene Werte sind `immediate` und `pending-reboot`.

> **Note**
>
> Parameter mit dem `ApplyType` von `static` müssen über einen `ApplyMethod` von `pending-reboot` verfügen.

Example - Ändern eines Parameterwerts

In diesem Beispiel listen Sie die Parameterwerte von `sample-parameter-group` auf und ändern den `tls`-Parameter. Anschließend listen Sie nach einer Wartezeit von 5 Minuten die Parameterwerte von `sample-parameter-group` erneut auf, um die geänderten Parameterwerte anzuzeigen.

1. Listen Sie die Parameter von `sample-parameter-group` und ihre Werte auf.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-cluster-parameters \
          --db-cluster-parameter-group-name sample-parameter-group
   ```

   Für Windows:

   ```
   aws docdb describe-db-cluster-parameters ^
          --db-cluster-parameter-group-name sample-parameter-group
   ```

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

   ```
   {
          "Parameters": [
              {
                  "Source": "system",
   ```

```
                "ApplyType": "static",
                "AllowedValues": "disabled,enabled",
                "ParameterValue": "enabled",
                "ApplyMethod": "pending-reboot",
                "DataType": "string",
                "ParameterName": "tls",
                "IsModifiable": true,
                "Description": "Config to enable/disable TLS"
            },
            {
                "Source": "user",
                "ApplyType": "dynamic",
                "AllowedValues": "disabled,enabled",
                "ParameterValue": "enabled",
                "ApplyMethod": "pending-reboot",
                "DataType": "string",
                "ParameterName": "ttl_monitor",
                "IsModifiable": true,
                "Description": "Enables TTL Monitoring"
            }
        ]
}
```

2. Ändern Sie den Parameter `tls`, sodass der Wert `disabled` ist.

   Sie können den `ApplyMethod` nicht ändern, da der `ApplyType` `static` ist.

   Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --parameters
 "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
```

   Für Windows:

```
aws docdb modify-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --parameters
 "ParameterName"=tls,"ParameterValue"=disabled,"ApplyMethod"=pending-reboot
```

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

```
{
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
```

3. Warten Sie mindestens 5 Minuten.
4. Listen Sie die Parameterwerte von `sample-parameter-group` auf, um zu überprüfen, ob der Parameter `tls` geändert wurde.

   Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
```

   Für Windows:

```
aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
```

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

```
{
       "Parameters": [
           {
               "ParameterValue": "false",
               "ParameterName": "enable_audit_logs",
               "ApplyType": "dynamic",
               "DataType": "string",
               "Description": "Enables auditing on cluster.",
               "AllowedValues": "true,false",
               "Source": "system",
               "IsModifiable": true,
               "ApplyMethod": "pending-reboot"
           },
           {
               "ParameterValue": "disabled",
               "ParameterName": "tls",
               "ApplyType": "static",
               "DataType": "string",
               "Description": "Config to enable/disable TLS",
               "AllowedValues": "disabled,enabled",
```

```
                "Source": "system",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            }
        ]
}
```

Ändern von Amazon DocumentDB-Clustern zur Verwendung benutzerdefinierter Cluster-Parametergruppen

Wenn Sie einen Amazon DocumentDB-Cluster erstellen, wird automatisch eine `default.docdb4.0` Parametergruppe für diesen Cluster erstellt. Sie können die Cluster-Parametergruppe `default` nicht ändern. Stattdessen können Sie Ihren Amazon DocumentDB-Cluster ändern, um ihm eine neue benutzerdefinierte Parametergruppe zuzuordnen.

In diesem Abschnitt wird erklärt, wie ein vorhandener Amazon DocumentDB-Cluster so geändert wird, dass er eine benutzerdefinierte Cluster-Parametergruppe mit AWS Management Console und AWS Command Line Interface (AWS CLI) verwendet.

Using the AWS Management Console

Um einen Amazon DocumentDB-Cluster so zu ändern, dass er eine neue, nicht standardmäßige Cluster-Parametergruppe verwendet

1. Bevor Sie beginnen, stellen Sie sicher, dass Sie einen Amazon DocumentDB-Cluster und eine Cluster-Parametergruppe erstellt haben. Weitere Anweisungen finden Sie unter Einen Amazon DocumentDB-Cluster erstellen und Amazon DocumentDB-Cluster-Parametergruppen erstellen.
2. Nachdem Sie Ihre Cluster-Parametergruppe erstellt haben, öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unterhttps://console.aws.amazon.com/docdb. Wählen Sie im Navigationsbereich Clusters (Cluster) aus, um die neue Parametergruppe einem Cluster hinzuzufügen.
3. Wählen Sie den Cluster aus, dem Sie die Parametergruppe zuordnen möchten. Wählen Sie Actions (Aktionen= und anschließend Modify (Ändern) aus, um den Cluster zu ändern.
4. Wählen Sie unter Cluster options (Clusteroptionen) die neue Parametergruppe aus, der Sie den Cluster zuordnen möchten.
5. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.

6. Nachdem Sie Ihre Änderungen überprüft haben, können Sie diese sofort oder während des nächsten Wartungsfensters unter Scheduling of modifications (Planen von Änderungen) anwenden.
7. Wählen Sie Modify cluster (Cluster ändern) aus, um den Cluster mit der neuen Parametergruppe zu aktualisieren.

Using the AWS CLI

Bevor Sie beginnen, stellen Sie sicher, dass Sie einen Amazon DocumentDB-Cluster und eine Cluster-Parametergruppe erstellt haben. Mit diesem AWS CLI `create-db-cluster` Vorgang können Sie einen Amazon DocumentDB-Cluster erstellen. Sie können mithilfe des AWS CLI `create-db-cluster-parameter-group` Vorgangs eine Cluster-Parametergruppe erstellen.

Verwenden Sie den AWS CLI `modify-db-cluster` Vorgang mit den folgenden Parametern, um Ihrem Cluster Ihre neue Cluster-Parametergruppe hinzuzufügen.

- -- db-cluster-identifier — Der Name Ihres Clusters (z. B.`sample-cluster`).
- -- db-cluster-parameter-group -name — Der Name der Parametergruppe, der Sie Ihren Cluster zuordnen möchten (z. B.`sample-parameter-group`).

Example

```
aws docdb modify-db-cluster \
    --db-cluster-identifier sample-cluster \
    --db-cluster-parameter-group-name sample-parameter-group
```

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

```
"DBCluster": {
    "AvailabilityZones": [
            "us-west-2c",
            "us-west-2b",
            "us-west-2a"
    ],
    "BackupRetentionPeriod": 1,
    "DBClusterIdentifier": "sample-cluster",
    "DBClusterParameterGroup": "sample-parameter-group",
    "DBSubnetGroup": "default",
```

```
    ...
}
```

Amazon DocumentDB-Cluster-Parametergruppen kopieren

Sie können eine Kopie einer Cluster-Parametergruppe in Amazon DocumentDB mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) erstellen.

Using the AWS Management Console

Das folgende Verfahren führt Sie durch die Erstellung einer neuen Cluster-Parametergruppe, indem Sie eine vorhandene Cluster-Parametergruppe kopieren.

So kopieren Sie eine Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) den Namen der Cluster-Parametergruppe aus, die Sie kopieren möchten.
4. Wählen Sie Actions (Aktionen) und anschließend Copy (Kopieren) aus, um diese Parametergruppe zu kopieren.
5. Geben Sie in Copy options (Kopieroptionen) einen Namen und eine Beschreibung für die neue Cluster-Parametergruppe ein. Wählen Sie anschließend Copy (Kopieren) aus, um Ihre Änderungen zu speichern.

Using the AWS CLI

Um eine Kopie einer Cluster-Parametergruppe zu erstellen, verwenden Sie die `copy-db-cluster-parameter-group`-Operation mit den folgenden Parametern.

- **`--source-db-cluster-parameter-group-identifier`** – Erforderlich. Der Name oder Amazon-Ressourcenname (ARN) der Cluster-Parametergruppe, von der Sie eine Kopie erstellen möchten.

  Wenn sich die Quell- und Zielcluster-Parametergruppen in derselben Gruppe befinden AWS-Region, kann der Identifier entweder ein Name oder ein ARN sein.

Wenn sich die Quell- und Zielcluster-Parametergruppen unterscheiden AWS-Regionen, muss es sich bei dem Bezeichner um einen ARN handeln.

- **`--target-db-cluster-parameter-group-identifier`** – Erforderlich. Der Name oder ARN der Kopie der Cluster-Parametergruppe.

  Einschränkungen:

  - Kann nicht Null, leer oder negativ sein.
  - Muss 1—255 Buchstaben, Zahlen oder Bindestriche enthalten.
  - Muss mit einem Buchstaben beginnen.
  - Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.
- **`--target-db-cluster-parameter-group-description`** – Erforderlich. Eine vom Benutzer angegebene Beschreibung für die Kopie der Cluster-Parametergruppe.

Example

Der folgende Code erstellt eine Kopie von `sample-parameter-group` und benennt die Kopie `sample-parameter-group-copy`.

Für Linux, macOS oder Unix:

```
aws docdb copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier sample-parameter-group \
    --target-db-cluster-parameter-group-identifier sample-parameter-group-copy \
    --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

Für Windows:

```
aws docdb copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier sample-parameter-group ^
    --target-db-cluster-parameter-group-identifier sample-parameter-group-copy ^
    --target-db-cluster-parameter-group-description "Copy of sample-parameter-group"
```

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

```
{
    "DBClusterParameterGroup": {
```

```
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:sample-parameter-group-copy",
        "DBClusterParameterGroupName": "sample-parameter-group-copy",
        "DBParameterGroupFamily": "docdb4.0",
        "Description": "Copy of sample-parameter-group"
    }
}
```

Amazon DocumentDB-Cluster-Parametergruppen zurücksetzen

Sie können einige oder alle Parameterwerte einer Amazon DocumentDB-Cluster-Parametergruppe auf ihre Standardwerte zurücksetzen, indem Sie das AWS Management Console oder das AWS Command Line Interface (AWS CLI) verwenden, um die Cluster-Parametergruppe zurückzusetzen.

Using the AWS Management Console

Führen Sie die folgenden Schritt aus, um einige oder alle Parameterwerte einer Cluster-Parametergruppe auf die Standardwerte zurückzusetzen.

So setzen Sie die Parameterwerte einer Cluster-Parametergruppe zurück

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Parameter groups (Parametergruppen) aus.
3. Wählen Sie im Bereich Cluster parameter groups (Cluster-Parametergruppen) den Namen der Cluster-Parametergruppe aus, die Sie zurücksetzen möchten.
4. Wählen Sie Actions (Aktionen) und anschließend Reset (Zurücksetzen) aus, um diese Parametergruppe zurückzusetzen.
5. Bestätigen Sie auf der Seite Cluster parameter group reset confirmation (Zurücksetzen der Cluster-Parametergruppe bestätigen), dass Sie alle Cluster-Parameter für diese Parametergruppe auf die Standardwerte zurücksetzen möchten. Wählen Sie anschließend Reset (Zurücksetzen) aus, um die Parametergruppe zurückzusetzen. Sie können auch Cancel (Abbrechen) auswählen, um die Änderungen zu verwerfen.

Using the AWS CLI

Um einige oder alle Parameterwerte einer Cluster-Parametergruppe auf ihre Standardwerte zurückzusetzen, verwenden Sie die `reset-db-cluster-parameter-group`-Operation mit den folgenden Parametern.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der zurückzusetzenden Cluster-Parametergruppe.
- **`--parameters`**— Fakultativ. Eine Liste von `ParameterName` und `ApplyMethod` in der Cluster-Parametergruppe, die auf ihre Standardwerte zurücksetzt werden sollen. Statische Parameter müssen auf `pending-reboot` gesetzt sein, damit sie beim nächsten Neustart der Instance oder `reboot-db-instance`-Anforderung wirksam werden. Sie müssen `reboot-db-instance` für alle Instances in Ihrem Cluster aufrufen, auf die der aktualisierte statische Parameter angewendet werden soll.

  Dieser Parameter und `--reset-all-parameters` schließen sich gegenseitig aus. Sie können jede der beiden Optionen verwenden, jedoch nicht beide.
- **`--reset-all-parameters`**oder **`--no-reset-all-parameters`** — Fakultativ. Gibt an, ob alle Parameter (`--reset-all-parameters`) oder nur einige der Parameter (`--no-reset-all-parameters`) auf ihre Standardwerte zurückgesetzt werden sollen. Der `--reset-all-parameters`-Parameter und `--parameters` schließen sich gegenseitig aus. Sie können jede der beiden Optionen verwenden, jedoch nicht beide.

  Wenn Sie die gesamte Gruppe zurücksetzen, werden dynamische Parameter sofort aktualisiert. Statische Parameter werden auf `pending-reboot` festgelegt, damit sie beim nächsten Neustart der Instance oder bei der nächsten `reboot-db-instance`-Anforderung wirksam werden. Sie müssen `reboot-db-instance` für alle Instances in Ihrem Cluster aufrufen, auf die der aktualisierte statische Parameter angewendet werden soll.

Example

Beispiel 1: Zurücksetzen aller Parameter auf ihre Standardwerte

Der folgende Code setzt alle Parameter in der Cluster-Parametergruppe `sample-parameter-group` auf ihre Standardwerte zurück.

Für Linux, macOS oder Unix:

```
aws docdb reset-db-cluster-parameter-group \
```

```
        --db-cluster-parameter-group-name sample-parameter-group \
        --reset-all-parameters
```

Für Windows:

```
aws docdb reset-db-cluster-parameter-group ^
        --db-cluster-parameter-group-name sample-parameter-group ^
        --reset-all-parameters
```

Beispiel 2: Zurücksetzen von angegebenen Parametern auf ihre Standardwerte

Der folgende Code setzt den `tls`-Parameter in der Cluster-Parametergruppe `sample-parameter-group` auf seinen Standardwert zurück.

Für Linux, macOS oder Unix:

```
aws docdb reset-db-cluster-parameter-group \
        --db-cluster-parameter-group-name sample-parameter-group \
        --no-reset-all-parameters \
        --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

Für Windows:

```
aws docdb reset-db-cluster-parameter-group ^
        --db-cluster-parameter-group-name sample-parameter-group ^
        --no-reset-all-parameters ^
        --parameters ParameterName=tls,ApplyMethod=pending-reboot
```

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

```
{
        "DBClusterParameterGroupName": "sample-parameter-group"
   }
```

Neustarten einer Cluster-Instance

Bevor ein statischer Parameterwert geändert wird, muss die Cluster-Instance neu gestartet werden. Starten Sie alle Instances in Ihrem Cluster neu, auf die der aktualisierte statische Parameter angewendet werden soll.

Für Linux, macOS oder Unix:

```
aws docdb reboot-db-instance \
       --db-instance-identifier sample-cluster-instance
```

Für Windows:

```
aws docdb reboot-db-instance ^
       --db-instance-identifier sample-cluster-instance
```

Löschen von Amazon DocumentDB-Cluster-Parametergruppen

Sie können eine benutzerdefinierte Amazon DocumentDB-Cluster-Parametergruppe mit dem AWS Management Console oder dem AWS Command Line Interface (AWS CLI) löschen. Sie können die Cluster-Parametergruppe `default.docdb4.0` nicht löschen.

Using the AWS Management Console

So löschen Sie eine Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parametergruppen das Optionsfeld links neben der Cluster-Parametergruppe aus, die Sie löschen möchten.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie im Bestätigungsbereich Delete (Löschen) die Option Delete (Löschen) aus, um die Cluster-Parametergruppe zu löschen. Um die Cluster-Parametergruppe beizubehalten, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Um eine Cluster-Parametergruppe zu löschen, verwenden Sie die `delete-db-cluster-parameter-group`-Operation mit dem folgenden Parameter.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der zu löschenden Cluster-Parametergruppe. Dabei muss es sich um eine vorhandene Cluster-Parametergruppe handeln. Sie können die Cluster-Parametergruppe *`default.docdb4.0`* nicht löschen.

Example - Löschen einer Cluster-Parametergruppe

Das folgende Beispiel führt Sie durch die drei Schritte zum Löschen einer Cluster-Parametergruppe:

1. Ermitteln des Namens der Cluster-Parametergruppe, die Sie löschen möchten.
2. Löschen der angegebene Cluster-Parametergruppe.
3. Überprüfen, ob die Cluster-Parametergruppe gelöscht wurde.

1. Suchen Sie den Namen der Cluster-Parametergruppe, die Sie löschen möchten.

Der folgende Code listet die Namen aller Cluster-Parametergruppen auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Die Ausgabe der vorherigen Operation ist eine Liste der Namen der Cluster-Parametergruppen ähnlich der folgenden (JSON-Format).

```
[
       [
           "default.docdb4.0"
       ],
```

```
        [
            "sample-parameter-group"
        ],
        [
            "sample-parameter-group-copy"
        ]
    ]
```

2. Löschen Sie eine angegebene Cluster-Parametergruppe.

Der folgenden Code löscht die Cluster-Parametergruppe `sample-parameter-group-copy`.

Für Linux, macOS oder Unix:

```
aws docdb delete-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Für Windows:

```
aws docdb delete-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group-copy
```

Diese Operation erzeugt keine Ausgabe.

3. Überprüfen Sie, ob die angegebene Cluster-Parametergruppe gelöscht wurde.

Der folgende Code listet die Namen aller verbleibenden Cluster-Parametergruppen auf.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
       --query 'DBClusterParameterGroups[*].[DBClusterParameterGroupName]'
```

Die Ausgabe der obigen Operation ist eine Liste der Cluster-Parametergruppen, die der folgenden ähnelt (JSON-Format). Die Cluster-Parametergruppe, die Sie gerade gelöscht haben, sollte nicht in der Liste enthalten sein.

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

```
[
       [
           "default.docdb4.0"
       ],
       [
           "sample-parameter-group"
       ]
   ]
```

Referenz zu den Amazon DocumentDB-Clusterparametern

Wenn Sie einen dynamischen Parameter ändern und die Cluster-Parametergruppe speichern, wird diese Änderung sofort übernommen, ungeachtet der Einstellung von Apply immediately (Sofort anwenden). Wenn Sie einen statischen Parameter ändern und eine Cluster-Parametergruppe speichern, wird die Änderung des Parameters nach einem manuellen Neustart der Instance angewendet. Sie können eine Instance über die Amazon DocumentDB DocumentDB-Konsole oder durch expliziten Aufruf `reboot-db-instance` neu starten.

Die folgende Tabelle zeigt die Parameter, die für alle Instances in einem Amazon DocumentDB-Cluster gelten.

Amazon DocumentDB DocumentDB-Parameter auf Clusterebene

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`audit_logs`	disabled	aktiviert, deaktiviert, ddl, dml_read, dml_write, all, none	Ja	Dynamisch	String	Definiert, ob CloudWatch Amazon-Audit-Logs aktiviert sind. • **`enabled`**—CloudWatch Audit-

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						Logs sind aktiviert. • **disabled**— CloudWatch Audit-Logs sind deaktiviert. • **ddl**— Die Überwachung von DDL-Ereignissen ist aktiviert. • **dml_read**— Die Prüfung auf DML-Leseereignisse ist aktiviert. • **dml_write** — Die Überwachung von DML-Schre

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						ibereignissen ist aktiviert. • **all**— Das Auditing für alle Datenbankereignisse ist aktiviert. • **none**— Auditing ist deaktiviert.
`change_stream_log_retention_duration`	10800	3600-604800	Ja	Dynamisch	Ganzzahl	Definiert den Zeitraum (in Sekunden), den das Change Stream-Protokoll aufbewahrt und genutzt werden kann.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`default_collection_compression`	disabled	aktiviert, deaktiviert	Ja	Dynamisch	String	Definiert die Standardkomprimierungseinstellung für neue Sammlungen in einem Cluster • **`enabled`**— Die Komprimierung ist standardmäßig aktiviert. • **`disabled`**— Die Komprimierung ist standardmäßig deaktiviert.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`profiler`	disabled	aktiviert, deaktiviert	Ja	Dynamisch	String	Aktiviert die Profilerstellung für langsame Operationen. • **`enabled`**— Vorgänge, die länger dauern als ein vom Kunden definierter Schwellenwert (z. B. 100 ms), werden in Amazon Logs protokolliert. CloudWatch • **`disabled`**— Langsame Operationen

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						werden nicht in Logs protokolliert. CloudWatch
`profiler_sampling_rate`	1,0	0,0 – 1,0	Ja	Dynamisch	Gleitkommazahl	Definiert die Sampling-Rate für protokollierte Operationen.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`profiler_threshold_ms`	100	50 – 2147483646	Ja	Dynamisch	Ganzzahl	Definiert den Schwellenwert für `profiler`. • Alle Operationen, die größer als `profiler_threshold_ms` sind, werden in CloudWatch Logs protokolliert.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`tls`	aktiviert	aktiviert, deaktiviert, fips-140-3, tls1.2+, tls1.3+	Ja	Statisch	String	Gibt an, ob TLS-Verbindungen (Transport Layer Security) erforderlich sind. • **`enabled`**— Für die Verbindung sind TLS-Verbindungen erforderlich. • **`disabled`**— TLS-Verbindungen können nicht für die Verbindung verwendet werden. • **`fips-140-3`** — Für die Verbindun

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						g sind TLS-Verbindungen mit FIPS-Attributen (Federal Information Processing Standards) erforderlich. Der Cluster akzeptiert nur sichere Verbindungen gemäß FIPS-Publikation 140-3. Dies wird erst ab Amazon DocumentDB 5.0-Clustern (Engine-

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						Version 3.0.3727) in diesen Regionen unterstützt: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west • **tls1.2+**— TLS-Verbindungen, die TLS Version 1.2 und höher verwenden, sind für die Verbindung erforderlich. Dies wird erst ab

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						Amazon DocumentDB 4.0 (Engine-Version 2.0.10980) und Amazon DocumentDB 5.0 (Engine-Version 3.0.11051) unterstützt. • **`tls1.3+`**— TLS-Verbindungen, die TLS Version 1.3 und höher verwenden, sind für die Verbindung erforderlich. Dies wird erst ab Amazon

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
						DocumentDB 4.0 (Engine-Version 2.0.10980) und Amazon DocumentDB 5.0 (Engine-Version 3.0.11051) unterstützt.

Parameter	Standardwert	Zulässige Werte	Anpassbar	Typ anwenden	Datentyp	Beschreibung
`ttl_monitor`	aktiviert	aktiviert, deaktiviert	Ja	Dynamisch	String	Gibt an, ob TTL-Überwachung (Time to Live) für den Cluster aktiviert ist. • **`enabled`**— Die TTL-Überwachung ist aktiviert. • **`disabled`**— Die TTL-Überwachung ist deaktiviert.

Amazon DocumentDB-Cluster-Parameter ändern

In Amazon DocumentDB bestehen Cluster-Parametergruppen aus Parametern, die für alle Instances gelten, die Sie im Cluster erstellen. Im Fall benutzerdefinierter Cluster-Parametergruppen können Sie für alle von Ihnen erstellten Parametergruppen die Parameterwerte jederzeit ändern oder alle Parameterwerte jederzeit auf die Standardwerte für Parametergruppen zurücksetzen. In diesem Abschnitt wird beschrieben, wie Sie die Parameter, aus denen eine Amazon DocumentDB-Cluster-Parametergruppe besteht, und ihre Werte anzeigen und wie Sie diese Werte ändern oder aktualisieren können.

Parameter können dynamisch oder statisch sein. Wenn Sie einen dynamischen Parameter ändern und die Cluster-Parametergruppe speichern, wird die Änderung sofort übernommen, ungeachtet der

Einstellung für `Apply Immediately`. Wenn Sie einen statischen Parameter ändern und die Cluster-Parametergruppe speichern, wird die Änderung des Parameters erst nach einem manuellen Neustart der Instances angewendet.

Parameter einer Amazon DocumentDB-Cluster-Parametergruppe anzeigen

Sie können die Parameter eines Amazon DocumentDB-Clusters und ihre Werte mit dem AWS Management Console oder AWS CLI sehen.

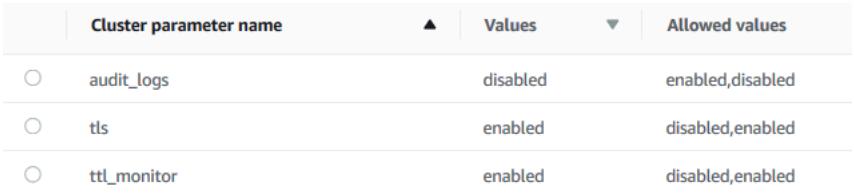
Using the AWS Management Console

So zeigen Sie die Details einer Cluster-Parametergruppe an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parameter groups (Parametergruppen) den Namen der Cluster-Parametergruppe aus, für die Sie die Details anzeigen möchten.
4. Auf der anschließend angezeigten Seite werden Ihnen für die einzelnen Parameter die folgenden Werte angezeigt: Name, aktueller Wert, zulässige Werte, ob der Parameter geändert werden kann, Anwendungstyp, Datentyp und Beschreibung.



Cluster parameter name	Values	Allowed values
audit_logs	disabled	enabled,disabled
tls	enabled	disabled,enabled
ttl_monitor	enabled	disabled,enabled

Using the AWS CLI

Um die Parameter einer Cluster-Parametergruppe und deren Werte anzuzeigen, verwenden Sie die `describe-db-cluster-parameters`-Operation mit den folgenden Parametern.

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, für die Sie eine detaillierte Parameterliste möchten.
- **`--source`**— Fakultativ. Wenn angegeben, werden nur Parameter für eine bestimmte Quelle zurückgegeben. Parameterquellen können `engine-default`, `system` oder `user` sein.

Example

Der folgende Code listet die Parameter und deren Werte für die Parametergruppe `custom3-6-param-grp` auf. Um weitere Informationen zur Parametergruppe zu erhalten, lassen Sie die Zeile `--query` aus. Um Informationen zu allen Parametergruppen zu erhalten, lassen Sie die Zeile `--db-cluster-parameter-group-name` aus.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
   --db-cluster-parameter-group-name custom3-6-param-grp \
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

Für Windows:

```
aws docdb describe-db-cluster-parameters ^
   --db-cluster-parameter-group-name custom3-6-param-grp ^
   --query 'Parameters[*].[ParameterName,ParameterValue]'
```

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

```
[
    [
        "audit_logs",
        "disabled"
    ],
    [
        "tls",
        "enabled"
    ],
    [
        "ttl_monitor",
        "enabled"
    ]
]
```

Parameter einer Amazon DocumentDB-Cluster-Parametergruppe ändern

Sie können die Parameter einer Parametergruppe mit dem AWS Management Console oder AWS CLIändern.

Using the AWS Management Console

So aktualisieren Sie die Parameter einer Cluster-Parametergruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Wählen Sie im Bereich Parameter groups (Parametergruppen) die Cluster-Parametergruppe aus, deren Parameter Sie aktualisieren möchten.
4. Auf der resultierenden Seite werden die Parameter und die entsprechenden Details zu dieser Cluster-Parametergruppe angezeigt. Wählen Sie einen zu aktualisierenden Parameter aus.
5. Klicken Sie oben rechts auf der Seite auf Edit (Bearbeiten), um den Wert des Parameters zu ändern. Weitere Informationen zu den Typen von Cluster-Parametern finden Sie unter Referenz zu den Amazon DocumentDB-Clusterparametern.
6. Führen Sie die Änderung aus. Wählen Sie anschließend Modify cluster parameter (Cluster-Parameter ändern) aus, um die Änderungen zu speichern. Um die Änderungen zu verwerfen, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Um die Parameter einer Cluster-Parametergruppe zu ändern, verwenden Sie die `modify-db-cluster-parameter-group`-Operation mit den folgenden Parametern:

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die Sie ändern.

- **`--parameters`** – Erforderlich. Der Parameter oder die Parameter, die Sie ändern. Jeder Parametereintrag muss Folgendes enthalten:
  - **`ParameterName`**— Der Name des Parameters, den Sie ändern.
  - **`ParameterValue`**— Der neue Wert für diesen Parameter.
  - **`ApplyMethod`**— Wie die Änderungen an diesem Parameter angewendet werden sollen. Zugelassene Werte sind `immediate` und `pending-reboot`.

    > **Note**
    >
    > Parameter mit dem `ApplyType` von `static` müssen über einen `ApplyMethod` von `pending-reboot` verfügen.

So ändern Sie die Werte der Parameter einer Cluster-Parametergruppe (AWS CLI)

Im folgenden Beispiel wird der Parameter `tls` geändert.

1. Listen Sie die Parameter von **`sample-parameter-group`** und ihre Werte auf.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-cluster-parameters \
       --db-cluster-parameter-group-name sample-parameter-group
   ```

   Für Windows:

   ```
   aws docdb describe-db-cluster-parameters ^
       --db-cluster-parameter-group-name sample-parameter-group
   ```

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

   ```
   {
       "Parameters": [
           {
               "Source": "system",
               "ApplyType": "static",
               "AllowedValues": "disabled,enabled",
               "ParameterValue": "enabled",
               "ApplyMethod": "pending-reboot",
   ```

```
            "DataType": "string",
            "ParameterName": "tls",
            "IsModifiable": true,
            "Description": "Config to enable/disable TLS"
        },
        {
            "Source": "user",
            "ApplyType": "dynamic",
            "AllowedValues": "disabled,enabled",
            "ParameterValue": "enabled",
            "ApplyMethod": "pending-reboot",
            "DataType": "string",
            "ParameterName": "ttl_monitor",
            "IsModifiable": true,
            "Description": "Enables TTL Monitoring"
        }
    ]
}
```

2. Ändern Sie den **`tls`**-Parameter, sodass sein Wert **`disabled`** ist. Sie können den `ApplyMethod` nicht ändern, da der `ApplyType` `static` ist.

   Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 "ParameterName"=tls,ParameterValue=disabled,ApplyMethod=pending-reboot"
```

   Für Windows:

```
aws docdb modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name sample-parameter-group ^
    --parameters "ParameterName=tls,ParameterValue=disabled,ApplyMethod=pending-
reboot"
```

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

```
{
    "DBClusterParameterGroupName": "sample-parameter-group"
}
```

3. Warten Sie mindestens 5 Minuten.
4. Listen Sie die Parameterwerte von **`sample-parameter-group`** auf.

   Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

   Für Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

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

```
{
    "Parameters": [
        {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "tls",
            "ParameterValue": "disabled",
            "Description": "Config to enable/disable TLS",
            "Source": "user",
            "ApplyType": "static",
            "DataType": "string",
            "AllowedValues": "disabled,enabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        }
    ]
}
```

Grundlegendes zu Amazon DocumentDB DocumentDB-Endpunkten

Sie können Amazon DocumentDB DocumentDB-Endpunkte (mit MongoDB-Kompatibilität) verwenden, um eine Verbindung zu einem Cluster oder einer Instance herzustellen. Amazon DocumentDB hat drei verschiedene Arten von Endpunkten, von denen jeder seinen eigenen Zweck hat.

Themen

- Die Endpunkte eines Clusters finden
- Den Endpunkt einer Instanz finden
- Verbindung zu Endpunkten herstellen

Cluster-Endpunkt

Ein Cluster-Endpunkt ist ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zur aktuellen primären Instance für den Cluster herstellt. Jeder Amazon DocumentDB-Cluster hat einen einzelnen Cluster-Endpunkt und eine primäre Instance. Im Falle eines Failovers wird der Cluster-Endpunkt auf die neue primäre Instance umgeschaltet.

Leser-Endpunkt

Ein Reader-Endpunkt ist ein Endpunkt für einen Amazon DocumentDB-Cluster, der eine Verbindung zu einem der verfügbaren Replikate für diesen Cluster herstellt. Jeder Amazon DocumentDB-Cluster hat einen Leser-Endpunkt. Wenn es mehr als ein Replikat gibt, leitet der Leser-Endpunkt jede Verbindungsanfrage an eines der Amazon DocumentDB DocumentDB-Replikate weiter.

Instance-Endpunkt

Ein Instance-Endpunkt ist ein Endpunkt, der sich mit einer bestimmten Instance verbindet. Jede Instance in einem Cluster, unabhängig davon, ob es sich um eine primäre oder eine Replikat-Instance handelt, hat ihren eigenen eindeutigen Instance-Endpunkt. Sie sollten keine Instance-Endpunkte in Ihrer Anwendung verwenden. Denn diese können im Falle eines Failovers Rollen ändern, sodass Code-Änderungen in Ihrer Anwendung erforderlich sind.

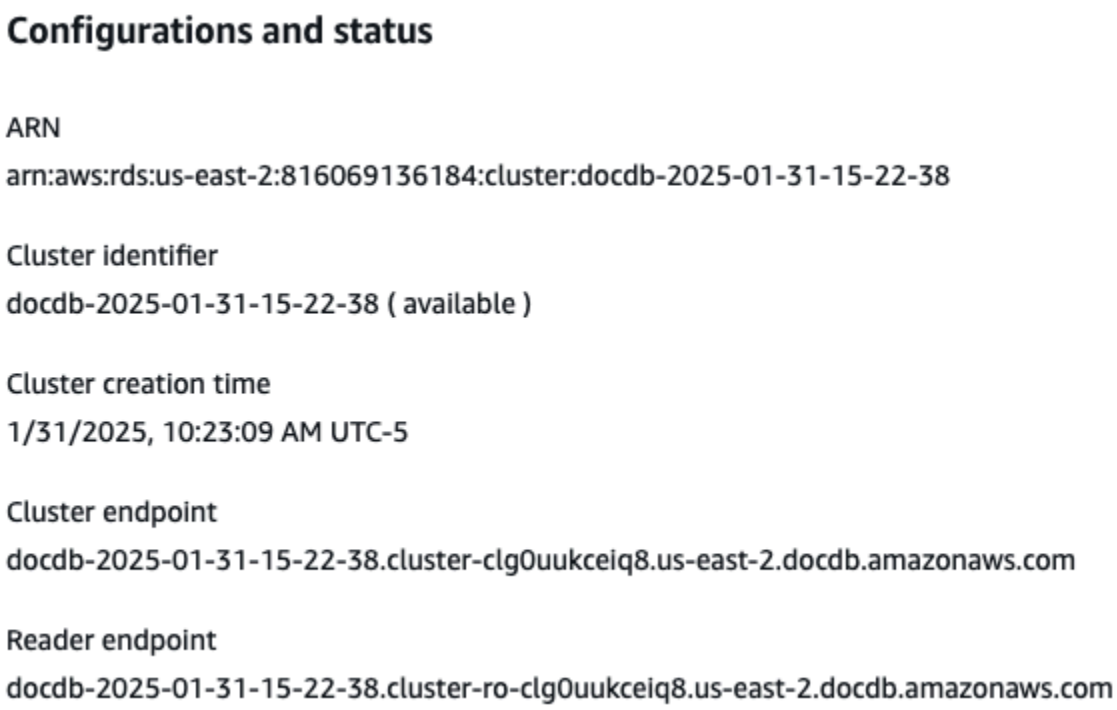
Die Endpunkte eines Clusters finden

Sie können den Cluster-Endpunkt und den Reader-Endpunkt eines Clusters mithilfe der Amazon DocumentDB DocumentDB-Konsole oder AWS CLI finden.

Using the AWS Management Console

Um die Endpunkte eines Clusters mithilfe der Konsole zu finden:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie aus der Liste der Cluster den Namen des Clusters, der Sie interessiert.
4. Wählen Sie auf der Seite mit den Cluster-Details die Registerkarte Konfiguration aus. Im Abschnitt Konfigurationen und Status finden Sie den Cluster-Endpunkt und den Reader-Endpunkt.



5. Um eine Verbindung zu diesem Cluster herzustellen, wählen Sie die Registerkarte Konnektivität und Sicherheit. Suchen Sie die Verbindungszeichenfolge für die `mongo` Shell und die Verbindungszeichenfolge, die im Anwendungscode verwendet werden kann, um eine Verbindung zu Ihrem Cluster herzustellen.



Using the AWS CLI

Um die Cluster- und Reader-Endpunkte für Ihren Cluster mithilfe von zu finden AWS CLI, führen Sie den `describe-db-clusters` Befehl mit diesen Parametern aus.

Parameter

- **`--db-cluster-identifier`**— Optional. Gibt den Cluster an, für den die Endpunkte zurückgegeben werden sollen. Wenn diese Option weggelassen wird, werden Endpunkte für bis zu 100 Ihrer Cluster zurückgegeben.
- **`--query`**—Fakultativ. Gibt die anzuzeigenden Felder an. Dies ist hilfreich, da die Datenmenge, die Sie anzeigen müssen, um die Endpunkte zu finden, reduziert wird. Wenn diese Option weggelassen wird, werden alle Informationen über einen Cluster zurückgegeben.
- **`--region`**—Fakultativ. Verwenden Sie den Parameter `--region`, um die Region anzugeben, für die Sie den Befehl anwenden möchten. Wenn diese Angabe weggelassen wird, wird Ihre Standardregion verwendet.

Example

Das folgende Beispiel liefert die Werte `DBClusterIdentifier`, Endpunkt (Cluster-Endpunkt) und `ReaderEndpoint` für `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
   --region us-east-1 \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
   --region us-east-1 ^
   --db-cluster-identifier sample-cluster ^
   --query 'DBClusters[*].[DBClusterIdentifier,Port,Endpoint,ReaderEndpoint]'
```

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

```
[
  [
     "sample-cluster",
     27017,
     "sample-cluster.cluster-corlsfccjozr.us-east-1.docdb.amazonaws.com",
     "sample-cluster.cluster-ro-corlsfccjozr.us-east-1.docdb.amazonaws.com"
  ]
]
```

Nachdem Sie nun den Cluster-Endpunkt haben, können Sie sich mit `mongo` oder `mongodb` mit dem Cluster verbinden. Weitere Informationen finden Sie unter Verbindung zu Endpunkten herstellen.

Den Endpunkt einer Instanz finden

Sie können den Endpunkt für eine Instance mithilfe der Amazon DocumentDB DocumentDB-Konsole oder der AWS CLI finden.

Using the AWS Management Console

So suchen Sie den Endpunkt einer Instance über die Konsole:

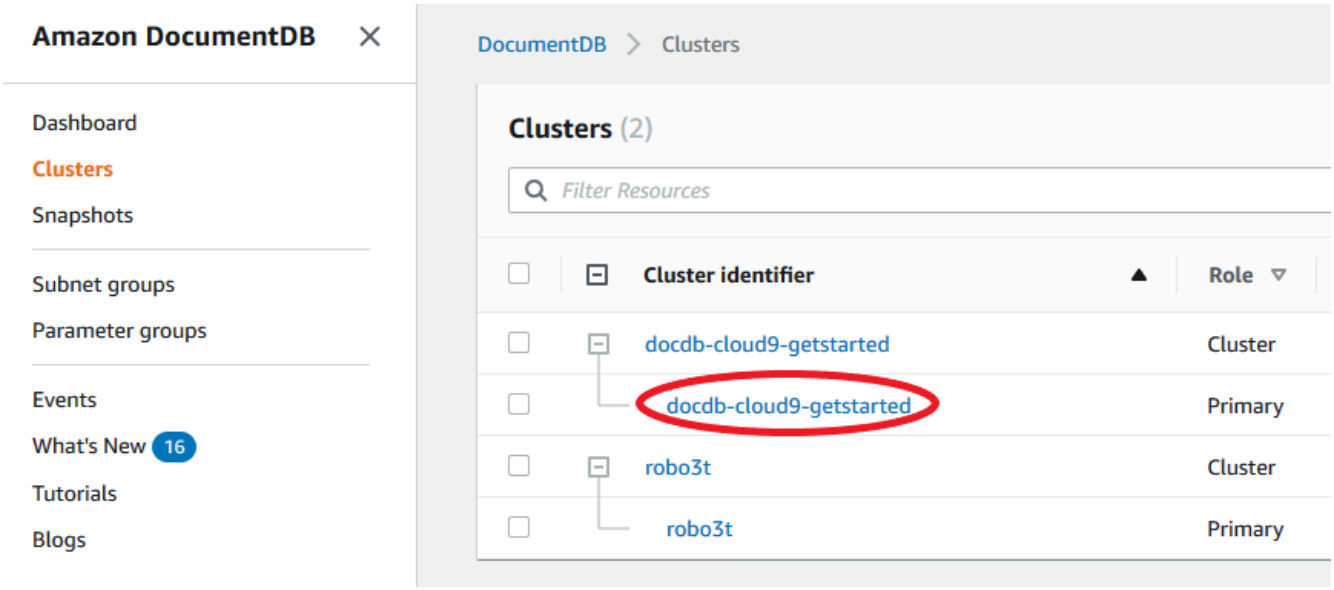
1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol

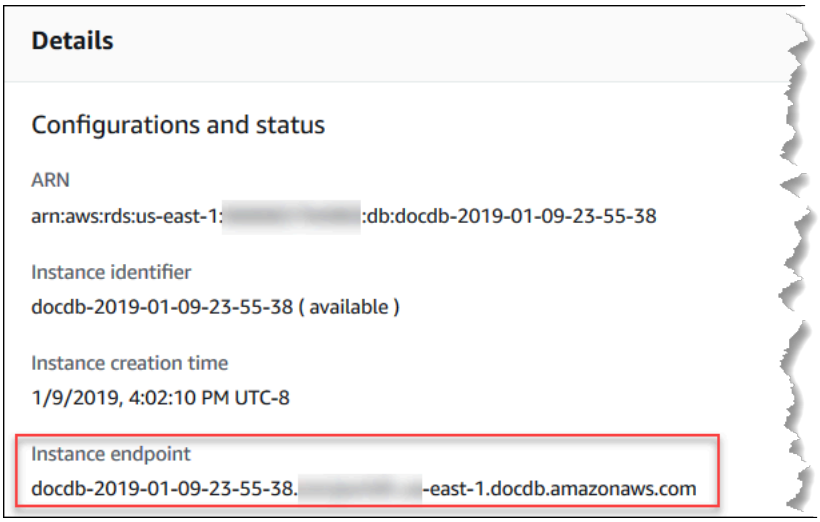
(≡)

aus.

3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Markieren Sie das Kästchen links neben der Instanz, an der Sie interessiert sind.
5. Scrollen Sie nach unten zum Abschnitt Details und suchen Sie dann den Instance-Endpunkt.



6. Um eine Verbindung zu dieser Instance herzustellen, scrollen Sie nach oben zum Abschnitt Connect (Verbinden). Suchen Sie die Verbindungszeichenfolge für die mongo-Shell und eine Verbindungszeichenfolge, die im Anwendungscode verwendet werden kann, um eine Verbindung zu Ihrer Instance herzustellen.



Using the AWS CLI

Führen Sie den folgenden Befehl mit diesen Argumenten aus AWS CLI, um den Instanzendpunkt mithilfe von zu finden.

Argumente

- **`--db-instance-identifier`**— Optional. Gibt die Instance an, für die der Endpunkt zurückgegeben werden soll. Wenn diese Option weggelassen wird, wird der Endpunkt für bis zu 100 Ihrer Instances zurückgegeben.
- **`--query`**—Fakultativ. Gibt die anzuzeigenden Felder an. Dies ist hilfreich, da die Datenmenge, die Sie anzeigen müssen, um die Endpunkte zu finden, reduziert wird. Wenn diese Option weggelassen wird, werden alle Informationen zu einer Instance zurückgegeben. Das Feld `Endpoint` hat drei Mitglieder, sodass die Auflistung in der Abfrage wie im folgenden Beispiel alle drei Mitglieder zurückgibt. Wenn Sie nur an einigen der `Endpoint`-Mitglieder interessiert sind, ersetzen Sie `Endpoint` in der Abfrage durch die Mitglieder, die Sie interessieren, wie im zweiten Beispiel.
- **`--region`**—Fakultativ. Verwenden Sie den Parameter `--region`, um die Region anzugeben, für die Sie den Befehl anwenden möchten. Wenn diese Angabe weggelassen wird, wird Ihre Standardregion verwendet.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

Für Windows:

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint]'
```

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

```
[
    [
        "sample-cluster-instance",
        {
            "Port": 27017,
            "Address": "sample-cluster-instance.corcjozrlsfc.us-
east-1.docdb.amazonaws.com",
            "HostedZoneId": "Z2R2ITUGPM61AM"
        }
    ]
]
```

Wenn Sie die Ausgabe reduzieren, um die `HostedZoneId` des Endpunkts zu eliminieren, können Sie Ihre Abfrage ändern, indem Sie `Endpoint.Port` und `Endpoint.Address` festlegen.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --region us-east-1 \
    --db-instance-identifier sample-cluster-instance \
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

Für Windows:

```
aws docdb describe-db-instances ^
    --region us-east-1 ^
    --db-instance-identifier sample-cluster-instance ^
    --query 'DBInstances[*].[DBInstanceIdentifier,Endpoint.Port,Endpoint.Address]'
```

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

```
[
    [
        "sample-cluster-instance",
```

```
        27017,
        "sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com"
    ]
]
```

Nachdem Sie nun den Instance-Endpunkt haben, können Sie sich mit mongo oder mongodb mit der Instance verbinden. Weitere Informationen finden Sie unter Verbindung zu Endpunkten herstellen.

Verbindung zu Endpunkten herstellen

Wenn Sie Ihren Endpunkt, entweder Cluster oder Instance, haben, können Sie sich über die mongo-Shell oder eine Verbindungszeichenfolge mit ihm verbinden.

Verbindung über die Mongo-Shell herstellen

Verwenden Sie die folgende Struktur, um die Zeichenfolge zu erstellen, die Sie für die Verbindung zu Ihrem Cluster oder Ihrer Instance über die mongo-Shell benötigen:

```
mongo \
    --ssl \
    --host Endpoint:Port \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

mongo-Shell-Beispiele

Verbinden mit einem Cluster:

```
mongo \
    --ssl \
    --host sample-cluster.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Verbindung zu einer Instance herstellen:

```
mongo \
    --ssl \
```

```
    --host sample-cluster-instance.corcjozrlsfc.us-east-1.docdb.amazonaws.com:27017 \
    --sslCAFile global-bundle.pem \
    --username UserName \
    --password Password
```

Verbindung über eine Verbindungszeichenfolge herstellen

Verwenden Sie die folgende Struktur, um die Verbindungszeichenfolge zu erstellen, die Sie für die Verbindung zu Ihrem Cluster oder Ihrer Instance benötigen.

```
mongodb://UserName:Password@endpoint:port?replicaSet=rs0&ssl_ca_certs=global-
bundle.pem
```

Beispiele für Verbindungszeichenfolgen

Verbinden mit einem Cluster:

```
mongodb://UserName:Password@sample-cluster.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Verbindung zu einer Instance herstellen:

```
mongodb://UserName:Password@sample-cluster-instance.cluster-corlsfccjozr.us-
east-1.docdb.amazonaws.com:27017?replicaSet=rs0&ssl_ca_certs=global-bundle.pem
```

Grundlegendes zu Amazon DocumentDB Amazon Resource Names () ARNs

Ressourcen, die Sie in erstellen, AWS werden jeweils eindeutig mit einem Amazon-Ressourcennamen (ARN) identifiziert. Für bestimmte Amazon DocumentDB DocumentDB-Operationen (mit MongoDB-Kompatibilität) müssen Sie eine Amazon DocumentDB DocumentDB-Ressource eindeutig identifizieren, indem Sie ihren ARN angeben. Wenn Sie beispielsweise ein Tag zu einer Ressource hinzufügen, müssen Sie den ARN der Ressource angeben.

Themen

- Einen ARN für eine Amazon DocumentDB DocumentDB-Ressource erstellen
- Einen ARN für eine Amazon DocumentDB DocumentDB-Ressource finden

Einen ARN für eine Amazon DocumentDB DocumentDB-Ressource erstellen

Sie können einen ARN für eine Amazon DocumentDB DocumentDB-Ressource mit der folgenden Syntax erstellen. Amazon DocumentDB verwendet das Format von Amazon Relational Database Service (Amazon RDS). ARNs Amazon DocumentDB ARNs enthält `rds` und nicht`docdb`.

`arn:aws:rds:`*`region`*`:`*`account_number`*`:`*`resource_type`*`:`*`resource_id`*

Name der Region	Region	Verfügbarkeitszonen (Datenverarbeitung)
USA Ost (Ohio)	`us-east-2`	3
USA Ost (Nord-Virginia)	`us-east-1`	6
USA West (Oregon)	`us-west-2`	4
Afrika (Kapstadt)	`af-south-1`	3
Südamerika (São Paulo)	`sa-east-1`	3
Asien-Pazifik (Hongkong)	`ap-east-1`	3
Asien-Pazifik (Hyderabad)	`ap-south-2`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	4
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3

Name der Region	Region	Verfügbarkeitszonen (Datenverarbeitung)
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Kanada (Zentral)	`ca-central-1`	3
Region China (Peking)	`cn-north-1`	3
China (Ningxia)	`cn-northwest-1`	3
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
Europa (Spain)	`eu-south-2`	3
Europa (Stockholm)	`eu-north-1`	3
Naher Osten (VAE)	`me-central-1`	3
Israel (Tel Aviv)	`il-central-1`	3
AWS GovCloud (USA West)	`us-gov-west-1`	3
AWS GovCloud (US-Ost)	`us-gov-east-1`	3

> **Note**
>
> Die Amazon DocumentDB DocumentDB-Architektur trennt Speicher und Datenverarbeitung. Für die Speicherebene repliziert Amazon DocumentDB sechs Kopien Ihrer Daten in drei AWS Availability Zones ()AZs. Die in der obigen Tabelle AZs aufgeführten sind die Anzahl der

Instances AZs , die Sie in einer bestimmten Region zur Bereitstellung von Compute-Instances verwenden können. Wenn Sie beispielsweise einen Amazon DocumentDB-Cluster in ap-northeast-1 starten, wird Ihr Speicher auf sechs Arten auf drei Arten repliziert, Ihre Compute-Instances sind AZs jedoch nur auf zwei verfügbar. AZs

Die folgende Tabelle zeigt das Format, das Sie verwenden sollten, wenn Sie einen ARN für eine bestimmte Amazon DocumentDB DocumentDB-Ressource erstellen. Amazon DocumentDB teilt das Format von Amazon RDS ARNs. Amazon DocumentDB ARNs enthält `rds` und nicht`docdb`.

Ressourcentyp	ARN-Format/Beispiel
Instance (db)	`arn:aws:rds:` *region*`:`*account_number* `:db:`*resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:db:`*sample-db-instance*
Cluster (`cluster`)	`arn:aws:rds:` *region*`:`*account_number* `:cluster:` *resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:cluster:` *sample-db-cluster*
Cluster-Parametergruppe (`cluster-pg`)	`arn:aws:rds:` *region*`:`*account_number* `:cluster-pg:` *resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:cluster-pg:` *sample-db-cluster-parameter-group*
Sicherheitsgruppe (`secgrp`)	`arn:aws:rds:` *region*`:`*account_number* `:secgrp:`*resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:secgrp:`*sample-public-secgrp*

Ressourcentyp	ARN-Format/Beispiel
Cluster-Snapshot (`cluster-snapshot`)	`arn:aws:rds:` *region*`:`*account_number* `:cluster-snapshot:` *resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:cluster-snapshot:` *sample-db-cluster-snapshot*
Subnetzgruppe (`subgrp`)	`arn:aws:rds:` *region*`:`*account_number* `:subgrp:`*resource_id* `arn:aws:rds:us-east-1:` *1234567890* `:subgrp:`*sample-subnet-10*

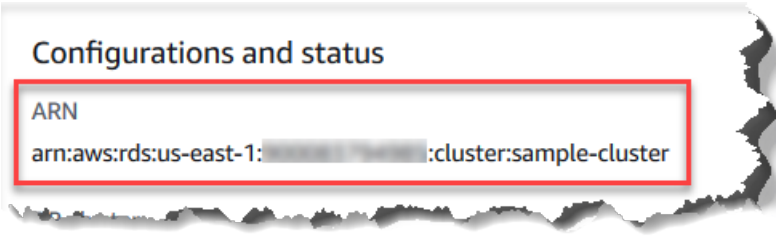
Einen ARN für eine Amazon DocumentDB DocumentDB-Ressource finden

Sie können den ARN einer Amazon DocumentDB DocumentDB-Ressource mit dem AWS Management Console oder dem AWS CLI ermitteln.

Using the AWS Management Console

Um einen ARN über die Konsole zu suchen, navigieren Sie zu der Ressource, für die Sie einen ARN erhalten möchten, und zeigen Sie die Details für diese Ressource an.

Sie können beispielsweise den ARN für einen Cluster abrufen, indem Sie auf der Cluster-Detailseite die Registerkarte Konfiguration auswählen. Die ARN finden Sie im Abschnitt Konfigurationen und Status, wie im folgenden Screenshot gezeigt.



Using the AWS CLI

Um einen ARN mit der AWS CLI für eine bestimmte Amazon DocumentDB DocumentDB-Ressource zu erhalten, verwenden Sie den `describe` Vorgang für diese Ressource. Die

folgende Tabelle zeigt jeden AWS CLI Vorgang und die ARN-Eigenschaft, die mit dem Vorgang zum Abrufen eines ARN verwendet wird.

AWS CLI Befehl	ARN-Eigenschaft
`describe-db-instances`	`DBInstanceArn`
`describe-db-clusters`	`DBClusterArn`
`describe-db-parameter-groups`	`DBParameterGroupArn`
`describe-db-cluster-parameter-groups`	`DBClusterParameterGroupArn`
`describe-db-security-groups`	`DBSecurityGroupArn`
`describe-db-snapshots`	`DBSnapshotArn`
`describe-db-cluster-snapshots`	`DBClusterSnapshotArn`
`describe-db-subnet-groups`	`DBSubnetGroupArn`

Example - ARN Ihres Clusters suchen

Mit der folgenden AWS CLI Operation wird der ARN für den Cluster ermitteltsample-cluster.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].DBClusterArn'
```

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

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster:sample-cluster"
]
```

Example - ARNs Suche nach mehreren Parametergruppen

Für Linux, macOS oder Unix:

```
aws docdb describe-db-cluster-parameter-groups \
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

Für Windows:

```
aws docdb describe-db-cluster-parameter-groups ^
   --query 'DBClusterParameterGroups[*].DBClusterParameterGroupArn'
```

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

```
[
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:custom3-6-param-grp",
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.aurora5.6",
    "arn:aws:rds:us-east-1:123456789012:cluster-pg:default.docdb3.6"
]
```

Taggen von Amazon DocumentDB DocumentDB-Ressourcen

Sie können Amazon DocumentDB-Tags (mit MongoDB-Kompatibilität) verwenden, um Metadaten zu Ihren Amazon DocumentDB DocumentDB-Ressourcen hinzuzufügen. Diese Tags können zusammen mit AWS Identity and Access Management (IAM-) Richtlinien verwendet werden, um den Zugriff auf Amazon DocumentDB DocumentDB-Ressourcen zu verwalten und zu kontrollieren, welche Aktionen auf die Ressourcen angewendet werden können. Sie können Tags auch verwenden, um Kosten zu verfolgen, indem Ausgaben für ähnlich markierte Ressourcen gruppiert werden.

Sie können die folgenden Amazon DocumentDB DocumentDB-Ressourcen taggen:

- Cluster
- Instances

- Snapshots
- Cluster-Snapshots
- Parametergruppen
- Cluster-Parametergruppen
- Sicherheitsgruppen
- Subnetzgruppen

Übersicht über Amazon DocumentDB DocumentDB-Ressourcen-Tags

Ein Amazon DocumentDB-Tag ist ein Name-Wert-Paar, das Sie definieren und mit einer Amazon DocumentDB DocumentDB-Ressource verknüpfen. Der Name wird als der Schlüssel bezeichnet. Die Angabe eines Wertes für den Schlüssel ist optional. Sie können Tags verwenden, um einer Amazon DocumentDB DocumentDB-Ressource beliebige Informationen zuzuweisen. Sie können einen Tag-Schlüssel z. B. dazu verwenden, um eine Kategorie zu definieren, und ein Tag-Wert könnte ein Element in dieser Kategorie sein. Sie könnten beispielsweise einen Tag-Schlüssel `project` und einen Tag-Wert von definieren`Salix`, was angibt, dass die Amazon DocumentDB DocumentDB-Ressource dem Salix-Projekt zugewiesen ist. Sie können Tags auch verwenden, um Amazon DocumentDB DocumentDB-Ressourcen so zu kennzeichnen, dass sie für Test- oder Produktionszwecke verwendet werden, indem Sie einen Schlüssel wie `environment=test` oder verwenden. `environment=production` Wir empfehlen Ihnen, einen konsistenten Satz von Tag-Schlüsseln zu verwenden, um die Nachverfolgung von Metadaten zu erleichtern, die Amazon DocumentDB DocumentDB-Ressourcen zugeordnet sind.

Sie können Tags verwenden, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Melden Sie sich dazu an, um Ihre AWS-Konto Rechnung mit den Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können. Weitere Informationen finden Sie unter Verwendung von Kostenzuordnungs-Tags im AWS Benutzerhandbuch für Fakturierung und Kostenmanagement.

Jede Amazon DocumentDB DocumentDB-Ressource hat einen Tagsatz, der alle Tags enthält, die dieser Ressource zugewiesen sind. Ein Tag-Satz kann bis zu zehn Tags enthalten oder leer sein. Wenn Sie einer Amazon DocumentDB DocumentDB-Ressource ein Tag hinzufügen, das denselben

Schlüssel wie ein vorhandenes Tag auf einer Ressource hat, überschreibt der neue Wert den alten Wert.

AWS weist Ihren Tags keine semantische Bedeutung zu; Tags werden ausschließlich als Zeichenketten interpretiert. Amazon DocumentDB kann Tags für eine Instance oder andere Amazon DocumentDB DocumentDB-Ressourcen festlegen, abhängig von den Einstellungen, die Sie bei der Erstellung der Ressource verwenden. Amazon DocumentDB könnte beispielsweise ein Tag hinzufügen, das angibt, dass eine Instance für die Produktion oder zum Testen bestimmt ist.

Sie können ein Tag an einen Snapshot anfügen. Diese Gruppierung taucht jedoch nicht in Ihrer Rechnung auf.

Sie können das AWS Management Console oder das verwenden, um Tags AWS CLI zu Amazon DocumentDB DocumentDB-Ressourcen hinzuzufügen, aufzulisten und zu löschen. Wenn Sie den verwenden AWS CLI, müssen Sie den Amazon-Ressourcennamen (ARN) für die Ressource angeben, mit der Sie arbeiten möchten. Weitere Informationen zu Amazon DocumentDB finden Sie ARNs unterGrundlegendes zu Amazon DocumentDB Amazon Resource Names () ARNs.

Tag-Einschränkungen

Die folgenden Einschränkungen gelten für Amazon DocumentDB-Tags:

- Maximale Anzahl von Tags pro Ressource: 10
- Maximale Schlüssellänge – 128 Unicode-Zeichen
- Maximale Wertlänge – 256 Unicode-Zeichen
- Gültige Zeichen für Schlüssel und Wert – Groß- und Kleinbuchstaben im UTF-8-Zeichensatz, Ziffern, Leerzeichen und die folgenden Zeichen: `_ . : / = + -` und `@` (Java-Regex: `"^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"`)
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Das Präfix `aws:` kann nicht für die Tag-Schlüssel oder Werte verwendet werden. Dieses Präfix ist für AWS reserviert.

Hinzufügen und Aktualisieren von Tags in einer Amazon DocumentDB DocumentDB-Ressource

Mit dem AWS Management Console oder dem können Sie einer Ressource bis zu 10 Tags hinzufügen. AWS CLI

Using the AWS Management Console

Der Prozess für das Hinzufügen eines Tags zu einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie das Tag hinzufügen. In diesem Beispiel fügen Sie ein Tag zu einem Cluster hinzu.

So fügen Sie Tags einem Cluster über die Konsole hinzu oder aktualisieren sie

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus.
3. Wählen Sie den Namen des Clusters aus, zu dem Sie Tags hinzufügen möchten.
4. Scrollen Sie nach unten zum Abschnitt Tags und wählen Sie anschließend Bearbeiten aus.
5. Für alle Tags, die Sie dieser Ressource hinzufügen möchten, gehen Sie wie folgt vor:
   a. Um ein neues Tag hinzuzufügen, geben Sie den Namen des Tags in das Feld Schlüssel ein. Um den Wert eines Tags zu ändern, suchen Sie den Namen des Tags in der Spalte Schlüssel.
   b. Um dem Tag einen neuen oder aktualisierten Wert zu geben, geben Sie einen Wert für das Tag in das Feld Wert ein.
   c. Wenn Sie mehrere Tags hinzufügen möchten, klicken Sie auf Hinzufügen. Andernfalls klicken Sie auf Speichern, sobald Sie fertig sind.

Using the AWS CLI

Der Prozess für das Hinzufügen eines Tags zu einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie die Tags hinzufügen. In diesem Beispiel fügen Sie drei Tags zu einem Cluster hinzu. Das zweite Tag, `key2`, hat keinen Wert.

Verwenden Sie den AWS CLI Vorgang `add-tags-to-resource` mit diesen Parametern.

Parameter

- **`--resource-name`**— Der ARN der Amazon DocumentDB DocumentDB-Ressource, der Sie Tags hinzufügen möchten.
- **`--tags`**—Eine Liste der Tags (Schlüssel-Wert-Paar), die Sie dieser Ressource hinzufügen möchten, im folgenden Format. `Key=`*`key-name`*`,Value=`*`tag-value`*

Example

Für Linux, macOS oder Unix:

```
aws docdb add-tags-to-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

Für Windows:

```
aws docdb add-tags-to-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tags Key=key1,Value=value1 Key=key2 Key=key3,Value=value3
```

Die `add-tags-to-resource`-Operation erzeugt keine Ausgabe. Um die Ergebnisse der Operation anzuzeigen, verwenden Sie die `list-tags-for-resource`-Operation.

Tags in einer Amazon DocumentDB DocumentDB-Ressource auflisten

Sie können das AWS Management Console oder das verwenden AWS CLI , um eine Liste der Tags für eine Amazon DocumentDB DocumentDB-Ressource zu erhalten.

Using the AWS Management Console

Der Prozess für das Auflisten von Tags in einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie das Tag hinzufügen. In diesem Beispiel listen Sie die Tags für einen Cluster auf.

So listen Sie die Tags in einem Cluster mithilfe der Konsole auf

1. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus.
3. Wählen Sie den Namen des Clusters aus, für den Sie Tags auflisten möchten.
4. Um eine Liste der Tags in dieser Ressource anzuzeigen, scrollen Sie nach unten zum Abschnitt Tags.

Using the AWS CLI

Der Prozess für das Auflisten der Tags in einer Ressource ist ähnlich, unabhängig davon, für welche Ressource Sie das Tag auflisten. In diesem Beispiel listen Sie die Tags in einem Cluster auf.

Verwenden Sie die AWS CLI Operation `list-tags-for-resource` mit diesen Parametern.

Parameter

- **`--resource-name`**—Erforderlich. Der ARN der Amazon DocumentDB DocumentDB-Ressource, für die Sie Tags auflisten möchten.

Example

Für Linux, macOS oder Unix:

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

Für Windows:

```
aws docdb list-tags-for-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster
```

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

```
{
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": ""
        },
        {
            "Key": "key3",
            "Value": "value3"
```

```
        }
    ]
}
```

Tags aus einer Amazon DocumentDB DocumentDB-Ressource entfernen

Sie können das AWS Management Console oder das verwenden, AWS CLI um Tags aus Amazon DocumentDB DocumentDB-Ressourcen zu entfernen.

Using the AWS Management Console

Der Prozess für das Entfernen von Tags aus einer Ressource ist ähnlich, unabhängig davon, welcher Ressource Sie das Tag hinzufügen. In diesem Beispiel entfernen Sie Tags aus einem Cluster.

So entfernen Sie Tags aus einem Cluster mithilfe der Konsole

1. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Cluster aus.
3. Wählen Sie den Namen des Clusters aus, aus dem Sie Tags entfernen möchten.
4. Scrollen Sie nach unten zum Abschnitt Tags und wählen Sie anschließend Bearbeiten aus.
5. Wenn Sie alle Tags aus dieser Ressource entfernen möchten, wählen Sie Remove all (Alle entfernen) aus. Andernfalls gehen Sie für alle Tags, die Sie aus dieser Ressource entfernen möchten, wie folgt vor:

    a. Suchen Sie den Namen des Tags in der Spalte Schlüssel.

    b. Wählen Sie Entfernen auf derselben Zeile wie der Tag-Schlüssel aus.

    c. Wenn Sie fertig sind, wählen Sie Speichern aus.

Using the AWS CLI

Der Prozess für das Entfernen eines Tags aus einer Ressource ist ähnlich, unabhängig davon, aus welcher Ressource Sie das Tag entfernen. In diesem Beispiel entfernen Sie ein Tag aus einem Cluster.

Verwenden Sie die AWS CLI Operation `remove-tags-from-resource` mit diesen Parametern.

- **`--resource-name`**—Erforderlich. Der ARN der Amazon DocumentDB DocumentDB-Ressource, aus der Sie Tags entfernen möchten.
- **`--tag-keys`**—Erforderlich. Eine Liste der Tag-Schlüssel, die von dieser Ressource entfernt werden sollen.

Example

Für Linux, macOS oder Unix:

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Für Windows:

```
aws docdb remove-tags-from-resource ^
    --resource-name arn:aws:rds:us-east-1:1234567890:cluster:sample-cluster \
    --tag-keys key1 key3
```

Die `removed-tags-from-resource`-Operation erzeugt keine Ausgabe. Um die Ergebnisse der Operation anzuzeigen, verwenden Sie die `list-tags-for-resource`-Operation.

Wartung von Amazon DocumentDB

Amazon DocumentDB führt regelmäßig Wartungsarbeiten an Amazon DocumentDB DocumentDB-Ressourcen durch. Diese Wartung umfasst in den meisten Fällen Aktualisierungen der Datenbank-Engine (Cluster-Wartung) oder des zugrunde liegenden Betriebssystems (OS) der Instance (Instance-Wartung). Datenbank-Engine-Updates sind erforderliche Patches und beinhalten Sicherheitskorrekturen, Bugfixes und Verbesserungen der Datenbank-Engine. Die meisten Betriebssystem-Patches sind zwar optional, aber wenn Sie sie eine Zeit lang nicht anwenden, kann es sein, dass der Patch erforderlich ist und auto angewendet wird, um Ihren Sicherheitsstatus aufrechtzuerhalten. Wir empfehlen Ihnen daher, Betriebssystem-Updates auf Ihre Amazon DocumentDB DocumentDB-Instances anzuwenden, sobald diese verfügbar sind.

Datenbank-Engine-Patches erfordern, dass Sie Ihre Amazon DocumentDB-Cluster für kurze Zeit offline schalten. Sobald sie verfügbar sind, werden diese Patches automatisch so geplant, dass sie während eines bevorstehenden geplanten Wartungsfensters Ihres Amazon DocumentDB-Clusters installiert werden.

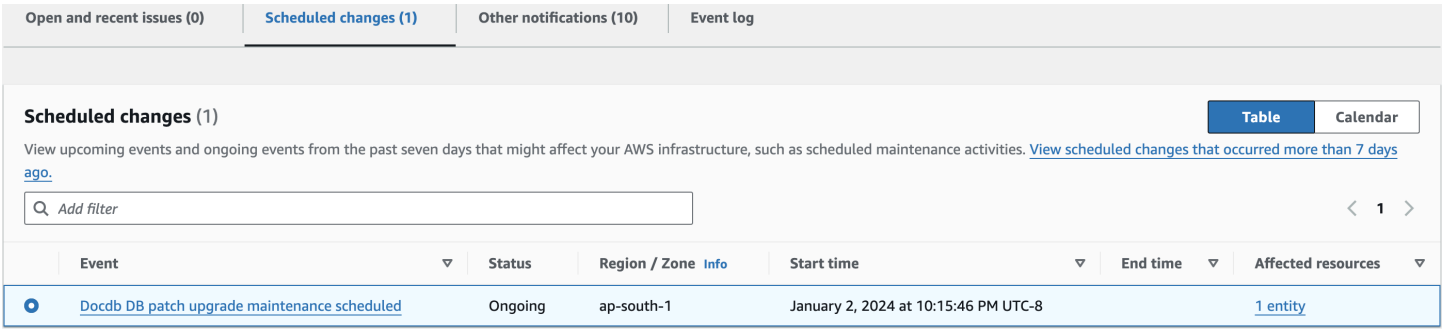
Sowohl die Cluster- als auch die Instance-Wartung haben jeweils eigene Wartungszeitfenster. Cluster- und Instance-Änderungen, die Sie nicht sofort anwenden möchten, werden auch während des Wartungsfensters angewendet. Wenn Sie einen Cluster erstellen, weist Amazon DocumentDB standardmäßig sowohl für einen Cluster als auch für jede einzelne Instance ein Wartungsfenster zu. Beim Anlegen eines Clusters oder einer Instance können Sie das Wartungsfenster auswählen. Sie können die Wartungsfenster außerdem jederzeit an Ihre Geschäftspläne oder -praktiken anpassen. Es wird generell empfohlen, Wartungsfenster auszuwählen, die die Auswirkungen der Wartung auf Ihre Anwendung minimieren (z. B. an Abenden oder Wochenenden).

Themen

- Benachrichtigungen für Amazon DocumentDB DocumentDB-Engine-Patches
- Ausstehende Amazon DocumentDB DocumentDB-Wartungsaktionen anzeigen
- Aktualisierungen der Amazon DocumentDB DocumentDB-Engine
- Vom Benutzer initiierte Updates
- Verwaltung Ihrer Amazon DocumentDB DocumentDB-Wartungsfenster
- Betriebssystem-Updates für Amazon DocumentDB

Benachrichtigungen für Amazon DocumentDB DocumentDB-Engine-Patches

Sie erhalten Wartungsbenachrichtigungen für erforderliche Datenbank-Engine-Patches über Integritätsereignisse im AWS Health Dashboard (AHD) in der AWS Konsole und per E-Mail. Wenn ein Wartungspatch für die Amazon DocumentDB DocumentDB-Engine in einer bestimmten AWS Region verfügbar wird, erhalten alle betroffenen Amazon DocumentDB DocumentDB-Benutzerkonten in der Region eine AHD- und E-Mail-Benachrichtigung für jede von dem Patch betroffene Amazon DocumentDB DocumentDB-Version. Sie können diese Benachrichtigungen im Bereich Geplante Änderungen des AHD in der Konsole einsehen. AWS Die Benachrichtigung enthält Informationen über den Zeitpunkt der Patch-Verfügbarkeit, den Zeitplan für die auto Anwendung, eine Liste der betroffenen Cluster und Versionshinweise. Diese Benachrichtigung wird auch per E-Mail an die E-Mail-Adresse des Root-Benutzers des AWS Kontos gesendet.



Sobald Sie diese Benachrichtigung erhalten haben, können Sie wählen, ob Sie diese Engine-Patches vor dem geplanten Datum für die automatische Anwendung selbst auf Ihre Amazon DocumentDB-Cluster anwenden möchten. Oder Sie können warten, bis die Engine-Patches während eines bevorstehenden Wartungsfensters automatisch angewendet werden (Standardoption).

> **Note**
>
> Der Status der Benachrichtigung im AHD wird auf „Laufend“ gesetzt, bis ein neuer Amazon DocumentDB DocumentDB-Engine-Patch mit einer neuen Engine-Patch-Version veröffentlicht wird.
> Sobald der Engine-Patch auf Ihren Amazon DocumentDB-Cluster angewendet wurde, wird die Engine-Patch-Version des Clusters entsprechend der Version in der Benachrichtigung aktualisiert. Sie können den `db.runCommand({getEngineVersion: 1})` Befehl ausführen, um dieses Update zu überprüfen.

AWS Health lässt sich auch in Amazon integrieren EventBridge , das Ereignisse verwendet, um skalierbare ereignisgesteuerte Anwendungen zu erstellen, und lässt sich in über 20 Ziele integrieren AWS Lambda, darunter Amazon Simple Queue Service (SQS) und andere. Sie können den `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_SCHEDULED` Ereigniscode verwenden, um Amazon einzurichten, EventBridge bevor Engine-Patches verfügbar werden. Sie können einrichten EventBridge , dass auf das Ereignis reagiert und automatisch Aktionen wie das Erfassen von Ereignisinformationen, das Auslösen zusätzlicher Ereignisse, das Senden von Benachrichtigungen über zusätzliche Kanäle wie Push-Benachrichtigungen an die und das Ergreifen von Abhilfemaßnahmen oder anderen Maßnahmen durchgeführt werden AWS Console Mobile Application, wenn ein Amazon DocumentDB DocumentDB-Engine-Patch in Ihrer Region verfügbar wird.

In dem seltenen Fall, dass Amazon DocumentDB einen Engine-Patch storniert, erhalten Sie eine AHD-Benachrichtigung sowie eine E-Mail, die Sie über die Kündigung informiert. Dementsprechend

können Sie den `AWS_DOCDB_DB_PATCH_UPGRADE_MAINTENANCE_CANCELLED` Ereigniscode verwenden, EventBridge um Amazon so einzurichten, dass es auf dieses Ereignis reagiert. Weitere Informationen zur Verwendung von EventBridge Amazon-Regeln finden Sie im EventBridge Amazon-Benutzerhandbuch.

Ausstehende Amazon DocumentDB DocumentDB-Wartungsaktionen anzeigen

Sie können mit dem AWS Management Console oder dem überprüfen, ob ein Wartungsupdate für Ihren Cluster verfügbar ist AWS CLI.

Wenn ein Update verfügbar ist, können Sie eine der folgenden Aktionen durchführen:

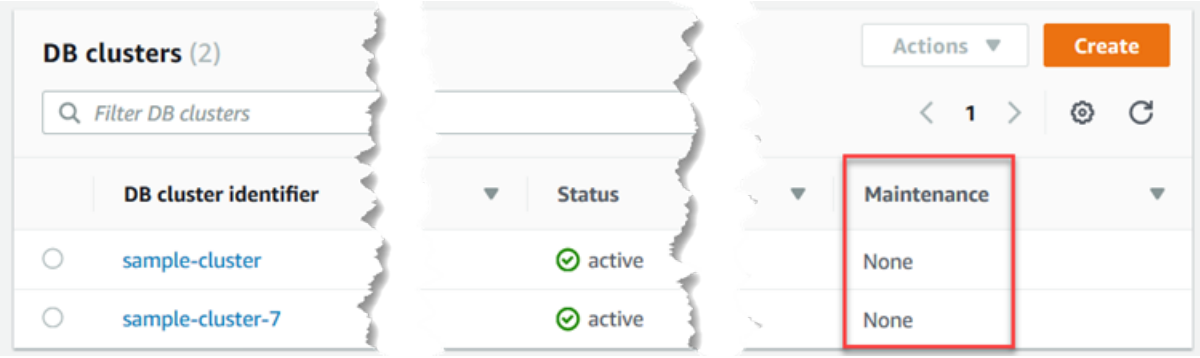
- Verschieben Sie eine Wartungsaktion, die derzeit für das nächste Wartungsfenster geplant ist (nur für Betriebssystem-Patches).
- Sofortiges Durchführen der Wartungsaktivitäten-
- Einplanen der Wartungsaktivitäten für das nächste Wartungsfenster.

> **Note**
>
> Wenn Sie keine Maßnahmen ergreifen, werden erforderliche Wartungsaktionen wie Engine-Patches in einem bevorstehenden geplanten Wartungsfenster auto angewendet.

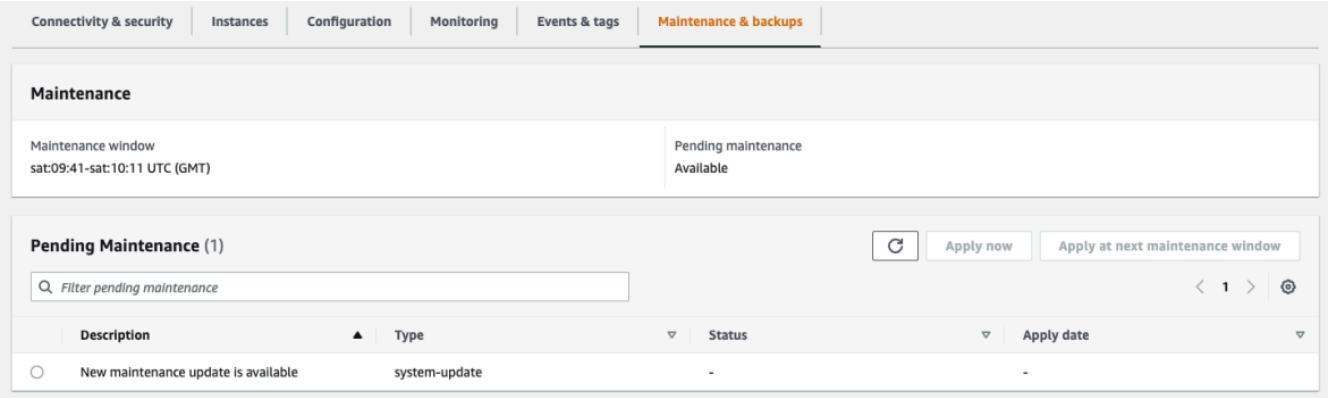
Das Wartungsfenster legt fest, wann die ausstehenden Operationen gestartet werden, gibt aber kein Abschlussdatum für diese Operationen vor; die Ausführungsdauer ist weder bekannt noch a priori beschränkt.

Using the AWS Management Console

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wenn ein Update verfügbar ist, wird es in der Spalte Wartung für den Cluster auf der Amazon DocumentDB DocumentDB-Konsole mit den Worten Verfügbar, Erforderlich oder Nächstes Fenster gekennzeichnet, wie hier dargestellt:



4. Um eine Maßnahme zu ergreifen, wählen Sie den Cluster aus, dessen Details angezeigt werden sollen, und wählen Sie dann Wartung und Backups. Die Einträge „Ausstehende Wartung“ werden angezeigt.



Using the AWS CLI

Verwenden Sie den folgenden AWS CLI Vorgang, um festzustellen, welche Wartungsmaßnahmen noch ausstehen. Die Ausgabe hier zeigt keine offenen Wartungsaktionen.

```
aws docdb describe-pending-maintenance-actions
```

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

```
{
    "PendingMaintenanceActions": []
}
```

Aktualisierungen der Amazon DocumentDB DocumentDB-Engine

Mit Amazon DocumentDB können Sie wählen, wann Wartungsarbeiten durchgeführt werden sollen. Mit dem AWS Management Console oder AWS CLI können Sie entscheiden, wann Amazon DocumentDB Updates einspielt.

Verwenden Sie die Verfahren in diesem Thema, um sofort ein Upgrade für Ihren Cluster durchzuführen oder ein Upgrade zu planen.

Using the AWS Management Console

Sie können die Konsole verwenden, um Updates für Ihre Amazon DocumentDB-Cluster zu verwalten.

Um ein Update für einen Cluster zu verwalten

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie in der Liste der Cluster die Schaltfläche neben dem Namen des Clusters, für den Sie den Wartungsvorgang anwenden möchten.
4. Wählen Sie aus dem Menü Actions (Aktionen) eine der folgenden Optionen aus:

   - Upgrade now (Upgrade jetzt), um die anstehenden Wartungsarbeiten sofort durchzuführen.
   - Upgrade at next window (Upgrade im nächsten Fenster), um die anstehenden Wartungsarbeiten im nächsten Wartungsfenster des Clusters durchzuführen.

   Alternativ können Sie im Bereich Ausstehende Wartung der Registerkarte Cluster-Wartung und Backups auf Jetzt anwenden oder Beim nächsten Wartungsfenster anwenden klicken (siehe Verwenden von AWS Management Console im vorherigen Abschnitt).

   > **Note**
   >
   > Wenn keine Wartungsaufgaben ausstehen, sind alle oben genannten Optionen inaktiv.

Using the AWS CLI

Verwenden Sie den `apply-pending-maintenance-action` AWS CLI Vorgang, um ein ausstehendes Update auf einen Cluster anzuwenden.

Parameter

- **`--resource-identifier`**— Der Amazon Resource Name (ARN) von Amazon DocumentDB der Ressource, für die die ausstehende Wartungsaktion gilt.
- **`--apply-action`**— Die ausstehende Wartungsaktion, die auf diese Ressource angewendet werden soll.

  Gültige Werte: `system-update` und `db-upgrade`.
- **`--opt-in-type`**— Ein Wert, der den Typ der Opt-in-Anfrage angibt oder eine Opt-in-Anfrage rückgängig macht. Eine Opt-in-Anfrage vom Typ `immediate` kann nicht rückgängig gemacht werden.

  Zulässige Werte:
  - `immediate`— Wendet die Wartungsmaßnahme sofort an.
  - `next-maintenance`— Wendet die Wartungsaktion im nächsten Wartungsfenster für die Ressource an.
  - `undo-opt-in`— Storniert alle bestehenden `next-maintenance` Opt-In-Anfragen.

Example

Für Linux, macOS oder Unix:

```
aws docdb apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb \
    --apply-action system-update \
    --opt-in-type immediate
```

Für Windows:

```
aws docdb apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:docdb ^
    --apply-action system-update ^
    --opt-in-type immediate
```

Verwenden Sie den Befehl, um eine Liste der Ressourcen zurückzugeben, für die `describe-pending-maintenance-actions` AWS CLI mindestens ein Update aussteht.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

Für Windows:

```
aws docdb describe-pending-maintenance-actions ^
    --resource-identifier arn:aws:rds:us-east-1:001234567890:db:docdb
```

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

```
{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-
east-1:001234567890:cluster:sample-cluster",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "system-update",
                    "CurrentApplyDate": "2019-01-11T03:01:00Z",
                    "Description": "db-version-upgrade",
                    "ForcedApplyDate": "2019-01-18T03:01:00Z",
                    "AutoAppliedAfterDate": "2019-01-11T03:01:00Z"
                }
            ]
        }
    ]
}
```

Sie können auch eine Liste von Ressourcen für einen Cluster zurückgeben, indem Sie den `--filters` Parameter des `describe-pending-maintenance-actions` AWS CLI Vorgangs angeben. Das Format für die `--filters`-Operation ist `Name=`*`filter-name`*`,Values=`*`resource-id`*`,...`.

`db-cluster-id`sind die akzeptablen Werte für den `Name` Parameter des Filters. Dieser Wert akzeptiert eine Liste von Cluster-Identifikatoren oder ARNs. Die zurückgegebene Liste enthält nur

ausstehende Wartungsaktionen für die Cluster, die durch diese Identifikatoren oder identifiziert wurden. ARNs

Das folgende Beispiel gibt die anstehenden Wartungsaktionen für die Cluster `sample-cluster1` und `sample-cluster2` zurück.

Example

Für Linux, macOS oder Unix:

```
aws docdb describe-pending-maintenance-actions \
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Für Windows:

```
aws docdb describe-pending-maintenance-actions ^
   --filters Name=db-cluster-id,Values=sample-cluster1,sample-cluster2
```

Termine anwenden

Jede Wartungsaktivität hat ein entsprechendes Anwendungsdatum, das Sie bei der Beschreibung der anstehenden Wartungsaktivität finden. Wenn Sie die Ausgabe der ausstehenden Wartungsmaßnahmen vom lesen AWS CLI, werden drei Daten aufgeführt. Diese Datumswerte geben an`null`, wann die Wartung optional ist. Die Werte werden aufgefüllt, sobald die entsprechende Wartungsaktion geplant oder durchgeführt wurde.

- **`CurrentApplyDate`**— Das Datum, an dem die Wartungsaktion entweder sofort oder während des nächsten Wartungsfensters angewendet wird.
- **`ForcedApplyDate`**— Das Datum, an dem die Wartung unabhängig von Ihrem Wartungsfenster automatisch angewendet wird.
- **`AutoAppliedAfterDate`**— Das Datum, nach dem die Wartung während des Wartungsfensters des Clusters angewendet wird.

Vom Benutzer initiierte Updates

Als Amazon DocumentDB DocumentDB-Benutzer können Sie Updates für Ihre Cluster oder Instances initiieren. Sie können beispielsweise die Klasse einer Instance in eine Klasse mit mehr oder weniger Speicher ändern, oder Sie können die Parametergruppe eines Clusters ändern. Amazon

DocumentDB betrachtet diese Änderungen anders als von Amazon DocumentDB initiierte Updates. Weitere Informationen zum Ändern eines Clusters oder einer Instance finden Sie in den folgenden Artikeln:

- Ändern eines Amazon DocumentDB-Clusters
- Ändern einer Amazon DocumentDB DocumentDB-Instance

Um eine Liste der ausstehenden, vom Benutzer initiierten Änderungen anzuzeigen, führen Sie den folgenden Befehl aus.

Example

So zeigen Sie ausstehende, von Benutzern initiierte Änderungen für Ihre Instances an:

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

Für Windows:

```
aws docdb describe-db-instances ^
    --query 'DBInstances[*].
[DBClusterIdentifier,DBInstanceIdentifier,PendingModifiedValues]'
```

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

In diesem Fall hat `sample-cluster-instance` eine ausstehende Änderung für eine `db.r5.xlarge` Instance-Klasse, während `sample-cluster-instance-2` keine ausstehenden Änderungen hat.

```
[
    [
        "sample-cluster",
        "sample-cluster-instance",
        {
            "DBInstanceClass": "db.r5.xlarge"
        }
    ],
    [
```

```
        "sample-cluster",
        "sample-cluster-instance-2",
        {}
    ]
]
```

Verwaltung Ihrer Amazon DocumentDB DocumentDB-Wartungsfenster

Jede Instance und jeder Cluster hat ein wöchentliches Wartungsfenster, in dem alle anstehenden Änderungen übernommen werden. Über das Wartungsfenster können Sie kontrollieren, wann angeforderte und erforderliche Änderungen und Software-Patches erfolgen. Wenn ein Wartungsereignis für eine bestimmte Woche geplant ist, wird es während des 30-minütigen Wartungsfensters eingeleitet, das Sie festlegen. Die meisten Wartungsereignisse werden auch während des 30-minütigen Wartungsfensters abgeschlossen, obwohl größere Wartungsereignisse länger als 30 Minuten dauern können.

Das 30-minütige Wartungsfenster wird zufällig aus einem 8-Stunden-Zeitraum pro Region ausgewählt. Wenn Sie bei der Erstellung der Instance oder des Clusters kein bevorzugtes Wartungsfenster angeben, weist Amazon DocumentDB an einem zufällig ausgewählten Wochentag ein 30-minütiges Wartungsfenster zu.

Die folgende Tabelle listet die Blöcke für jede Region auf, von denen Standard-Wartungsfenster zugewiesen werden.

Name der Region	Region	UTC-Zeitblock
USA Ost (Ohio)	us-east-2	03:00-11:00
USA Ost (Nord-Virginia)	us-east-1	03:00-11:00
USA West (Oregon)	us-west-2	06:00-14:00
Afrika (Kapstadt)	af-south-1	03:00 — 11:00
Asien-Pazifik (Hongkong)	ap-east-1	06:00-14:00
Asien-Pazifik (Hyderabad)	ap-south-2	06:30 — 14:30
Asien-Pazifik (Mumbai)	ap-south-1	06:00-14:00

Name der Region	Region	UTC-Zeitblock
Asien-Pazifik (Seoul)	ap-northeast-2	13:00-21:00
Asien-Pazifik (Singapur)	ap-southeast-1	14:00-22:00
Asien-Pazifik (Sydney)	ap-southeast-2	12:00-20:00
Asien-Pazifik (Tokio)	ap-northeast-1	13:00-21:00
Kanada (Zentral)	ca-central-1	03:00-11:00
China (Beijing)	cn-north-1	06:00-14:00
China (Ningxia)	cn-northwest-1	06:00-14:00
Europa (Frankfurt)	eu-central-1	21:00-05:00
Europa (Irland)	eu-west-1	22:00-06:00
Europa (London)	eu-west-2	22:00-06:00
Europa (Mailand)	eu-south-1	02:00-10:00
Europa (Paris)	eu-west-3	23:59-07:29
Europa (Spanien)	eu-south-2	02:00 — 10:00
Europa (Stockholm)	eu-north-1	04:00 — 12:00
Naher Osten (VAE)	me-central-1	05:00 — 13:00
Südamerika (São Paulo)	sa-east-1	00:00-08:00
Israel (Tel Aviv)	il-central-1	04:00-12:00
AWS GovCloud (US-Ost)	us-gov-east-1	17:00-01:00
AWS GovCloud (US-West)	us-gov-west-1	06:00-14:00

Ihre Amazon DocumentDB DocumentDB-Wartungsfenster ändern

Das Wartungsfenster sollte in den Zeitraum mit der geringsten Nutzung fallen und daher unter Umständen von Zeit zu Zeit geändert werden. Ihr Cluster oder Ihre Instance ist während dieser Zeit nur dann nicht verfügbar, wenn Systemänderungen (z. B. eine Speicherskalierung oder ein Instance-Klassen-Wechsel) durchgeführt werden und einen Ausfall erforderlich machen. Die fehlende Verfügbarkeit gilt dann nur für die minimale Zeitspanne, die für die notwendigen Änderungen benötigt wird.

Für Upgrades der Datenbank-Engine verwendet Amazon DocumentDB das bevorzugte Wartungsfenster des Clusters und nicht das Wartungsfenster für einzelne Instances.

So ändern Sie das Wartungsfenster

- Für einen Cluster: Weitere Informationen finden Sie unter Ändern eines Amazon DocumentDB-Clusters.
- Für eine Instance: Weitere Informationen finden Sie unter Ändern einer Amazon DocumentDB DocumentDB-Instance.

Betriebssystem-Updates für Amazon DocumentDB

Instances in Amazon DocumentDB-Clustern erfordern gelegentlich Betriebssystemupdates. Amazon DocumentDB aktualisiert das Betriebssystem auf eine neuere Version, um die Datenbankleistung und die allgemeine Sicherheitslage der Kunden zu verbessern. Betriebssystem-Updates ändern nicht die Cluster-Engine-Version oder Instance-Klasse einer Amazon DocumentDB DocumentDB-Instance.

Wir empfehlen, dass Sie zuerst die Reader-Instances in einem Cluster und dann die Writer-Instance aktualisieren, um die Verfügbarkeit Ihres Clusters zu maximieren. Es wird nicht empfohlen, die Reader- und Writer-Instances gleichzeitig zu aktualisieren, da es bei einem Failover zu längeren Ausfallzeiten kommen kann.

Die meisten Betriebssystemupdates für Amazon DocumentDB sind optional und haben kein festes Datum, an dem sie angewendet werden sollen. Wenn Sie diese Updates jedoch eine Zeit lang nicht installieren, werden sie möglicherweise irgendwann erforderlich und werden während des Wartungsfensters Ihrer Instance automatisch angewendet. Dies soll dazu beitragen, den Sicherheitsstatus Ihrer Datenbank aufrechtzuerhalten. Um unerwartete Ausfallzeiten zu vermeiden, empfehlen wir Ihnen, Betriebssystem-Updates auf Ihre Amazon DocumentDB DocumentDB-Instances anzuwenden, sobald diese verfügbar sind, und Ihr Instance-Wartungsfenster zu einem Zeitpunkt festzulegen, der Ihren Geschäftsanforderungen entspricht.

Wenn Sie benachrichtigt werden möchten, sobald eine neue optionale Aktualisierung verfügbar ist, können Sie RDS-EVENT-0230 in der Kategorie Sicherheitspatch-Ereignis abonnieren. Informationen zum Abonnieren von Amazon DocumentDB DocumentDB-Veranstaltungen finden Sie unter Amazon DocumentDB DocumentDB-Veranstaltungsabonnements abonnieren.

Sie sollten davon ausgehen, dass, wenn eine Wartung auf Ihrem Cluster oder Ihrer Instance ausgeführt wird, und es sich um eine primäre Instance handelt, ein Failover ausgeführt wird. Um Ihre Verfügbarkeit zu verbessern, empfehlen wir Ihnen, mehr als eine Instance für Ihre Amazon DocumentDB-Cluster zu verwenden. Weitere Informationen finden Sie unter Amazon DocumentDB-Failover.

> **Note**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird.

> **Important**
>
> Ihre Amazon DocumentDB DocumentDB-Instance wird während des Betriebssystem-Upgrades offline geschaltet. Sie können die Cluster-Ausfallzeiten minimieren, indem Sie einen Cluster mit mehreren Instanzen einrichten. Wenn Sie keinen Cluster mit mehreren Instanzen haben, können Sie vorübergehend einen erstellen, indem Sie sekundäre Instanzen hinzufügen, um diese Wartung durchzuführen, und dann die zusätzlichen Reader-Instances löschen, sobald die Wartung abgeschlossen ist (es fallen reguläre Gebühren für die sekundäre Instanz an).

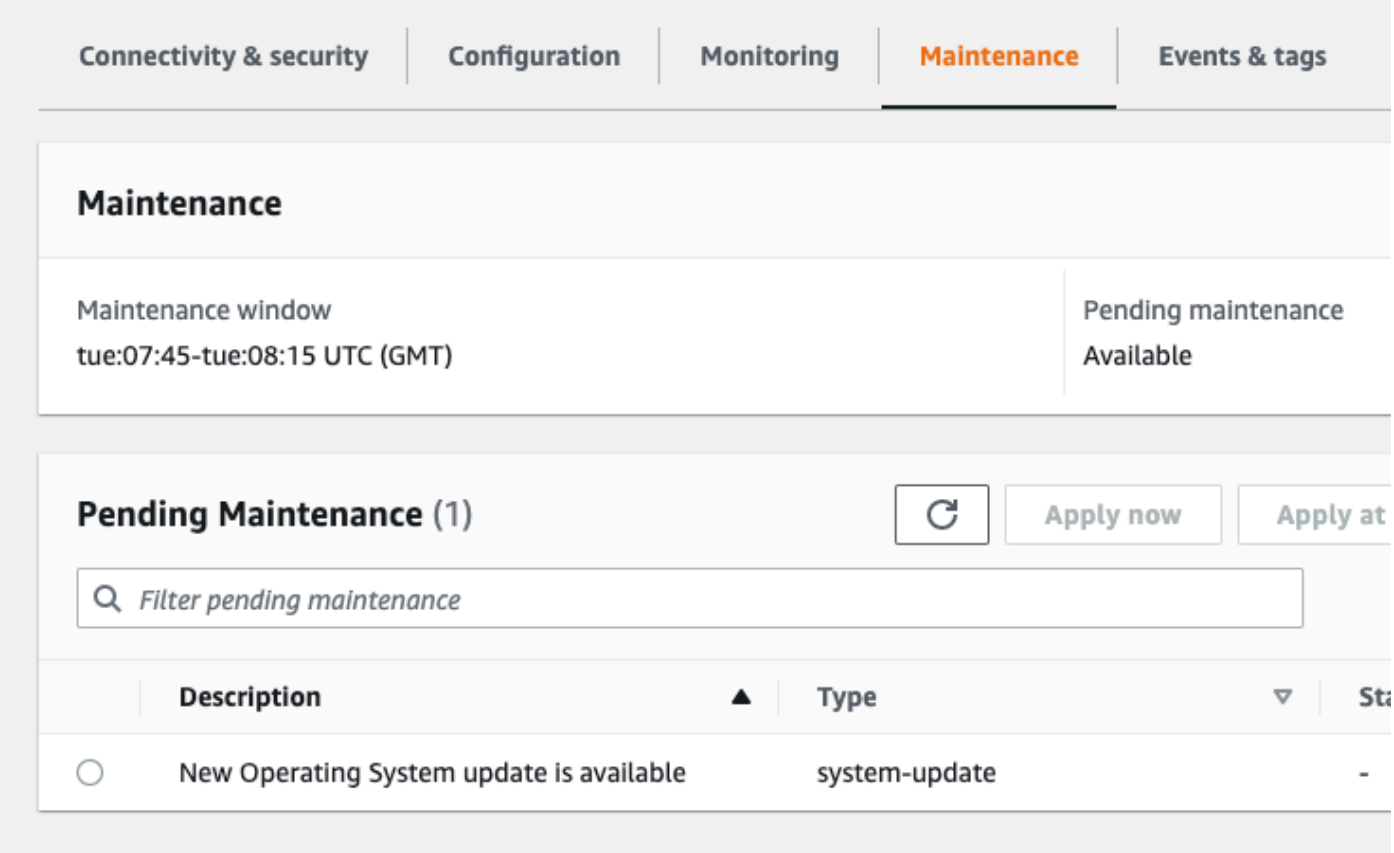
> **Note**
>
> Es ist möglicherweise erforderlich, im Hinblick auf alle optionalen und obligatorischen Updates auf dem Laufenden zu bleiben, um verschiedene Compliance-Auflagen zu erfüllen. Wir empfehlen, dass Sie alle von Amazon DocumentDB bereitgestellten Updates während Ihrer Wartungsfenster routinemäßig anwenden.

Sie können das AWS Management Console oder das verwenden AWS CLI , um festzustellen, ob ein Update verfügbar ist.

Using the AWS Management Console

Um festzustellen, ob ein Update verfügbar ist, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Clusters und dann die Instance aus.
3. Wählen Sie Maintenance aus.
4. Suchen Sie im Abschnitt Ausstehende Wartung nach dem Betriebssystem-Update.



Sie können das Betriebssystem-Update auswählen und im Abschnitt Ausstehende Wartung auf Jetzt anwenden oder Beim nächsten Wartungsfenster anwenden klicken. Wenn der Wartungswert „Nächstes Fenster“ lautet, verschieben Sie die Wartungsarbeiten, indem Sie „Upgrade aufschieben“ wählen. Sie können eine Wartungsaktion nicht verschieben, wenn sie bereits gestartet wurde.

Alternativ können Sie die Instanz aus einer Liste von Clustern auswählen, indem Sie im Navigationsbereich auf Cluster klicken und im Menü Aktionen die Option Jetzt anwenden oder Beim nächsten Wartungsfenster anwenden auswählen.

Using the AWS CLI

Rufen Sie den folgenden Befehl auf, um festzustellen, ob ein Update verfügbar ist AWS CLI, indem Sie den `describe-pending-maintenance-actions` folgenden Befehl aufrufen:

```
aws docdb describe-pending-maintenance-actions
```

```
{
  "ResourceIdentifier": "arn:aws:docdb:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Betriebssystem-Updates sind spezifisch für Amazon DocumentDB DocumentDB-Engine-Versionen und Instance-Klassen. Daher erhalten oder benötigen Amazon DocumentDB DocumentDB-Instances Updates zu unterschiedlichen Zeiten. Wenn für Ihre Instance je nach Engine-Version und Instance-Klasse ein Betriebssystem-Update verfügbar ist, wird das Update in der Konsole angezeigt. Es kann auch angezeigt werden, indem AWS CLI `describe-pending-maintenance-actions` Sie den Befehl ausführen oder den `DescribePendingMaintenanceActions` API-Vorgang aufrufen.

Wenn Sie nicht die neueste Cluster-Patch-Version Ihrer Amazon DocumentDB DocumentDB-Engine ausführen, wird das Betriebssystem-Update möglicherweise nicht als verfügbare Wartung aufgeführt. Um das Betriebssystem-Update anzeigen und verwalten zu können, sollten Sie zunächst ein Upgrade auf die neueste Engine-Patch-Version durchführen.

Grundlegendes zu Rollen im Zusammenhang mit Services

Amazon DocumentDB (mit MongoDB-Kompatibilität) verwendet AWS Identity and Access Management (IAM) service-verknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon DocumentDB verknüpft ist. Servicebezogene Rollen sind von

Amazon DocumentDB vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle erleichtert die Verwendung von Amazon DocumentDB, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon DocumentDB definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon DocumentDB seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dadurch werden Ihre Amazon DocumentDB DocumentDB-Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Services, die serviceverknüpften Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Mit dem Service verknüpfte Amazon DocumentDB DocumentDB-Rollenberechtigungen

Amazon DocumentDB (mit MongoDB-Kompatibilität) verwendet die serviceverknüpfte Rolle namens AWSServiceRoleForRDS, damit Amazon DocumentDB AWS Dienste im Namen Ihrer Cluster aufrufen kann.

Die dienstbezogene AWSService RoleFor RDS-Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `docdb.amazonaws.com`

Die Rollenberechtigungsrichtlinie ermöglicht es Amazon DocumentDB, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktionen auf `ec2`:
  - `AssignPrivateIpAddresses`
  - `AuthorizeSecurityGroupIngress`
  - `CreateNetworkInterface`

  - `CreateSecurityGroup`
  - `DeleteNetworkInterface`
  - `DeleteSecurityGroup`
  - `DescribeAvailabilityZones`
  - `DescribeInternetGateways`
  - `DescribeSecurityGroups`
  - `DescribeSubnets`
  - `DescribeVpcAttribute`
  - `DescribeVpcs`
  - `ModifyNetworkInterfaceAttribute`
  - `RevokeSecurityGroupIngress`
  - `UnassignPrivateIpAddresses`
- Aktionen auf `sns`:
  - `ListTopic`
  - `Publish`
- Aktionen auf `cloudwatch`:
  - `PutMetricData`
  - `GetMetricData`
  - `CreateLogStream`
  - `PullLogEvents`
  - `DescribeLogStreams`
  - `CreateLogGroup`

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Möglicherweise erhalten Sie die folgende Fehlermeldung:
> Unable to create the resource. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu erstellen. Andernfalls warten Sie und versuchen Sie es später noch einmal.

Wenn Sie diesen Fehler erhalten, stellen Sie sicher, dass Sie die folgenden Berechtigungen aktiviert haben:

```
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
AWSServiceRoleForRDS",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"rds.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Eine serviceverknüpfte Amazon DocumentDB DocumentDB-Rolle erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen Cluster erstellen, erstellt Amazon DocumentDB die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und dann erneut erstellen müssen, können Sie die Rolle in Ihrem Konto mit demselben Verfahren neu anlegen. Wenn Sie einen Cluster erstellen, erstellt Amazon DocumentDB die serviceverknüpfte Rolle erneut für Sie.

Änderung einer serviceverknüpften Amazon DocumentDB DocumentDB-Rolle

Amazon DocumentDB erlaubt es Ihnen nicht, die mit dem AWSService RoleFor RDS-Dienst verknüpfte Rolle zu ändern. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können die Beschreibung der Rolle jedoch mithilfe von IAM ändern. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Amazon DocumentDB DocumentDB-Rolle

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 alle Ihre -Cluster löschen, bevor Sie die serviceverknüpfte Rolle löschen können.

Bereinigen einer serviceverknüpften Rolle mit Amazon DocumentDB

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie über die Konsole, ob die servicegebundene Rolle eine aktive Sitzung hat:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im Navigationsbereich der IAM-Konsole Rollen und dann den Namen (nicht das Kontrollkästchen) der AWSServiceRoleForRDS-Rolle aus.
3. Wählen Sie auf der Seite Summary für die ausgewählte Rolle die Registerkarte Access Advisor.
4. Überprüfen Sie auf der Registerkarte Access Advisor (Advisor aufrufen) die jüngsten Aktivitäten für die serviceverknüpfte Rolle.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob Amazon DocumentDB die AWSService RoleFor RDS-Rolle verwendet, können Sie versuchen, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt die Löschung fehl und Sie können die -Regionen anzeigen, in denen die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Wenn Sie die AWSService RoleFor RDS-Rolle entfernen möchten, müssen Sie zuerst alle Ihre Instances und Cluster löschen. Informationen zum Löschen von Instances und Clustern finden Sie in den folgenden Themen:

- Löschen einer Amazon DocumentDB DocumentDB-Instance
- Löschen eines Amazon DocumentDB-Clusters

Unterstützte Regionen für serviceverknüpfte Amazon DocumentDB DocumentDB-Rollen

Amazon DocumentDB unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html#regions-and-azs-availability.

Change-Streams mit Amazon DocumentDB verwenden

Die Change-Streams-Funktion in Amazon DocumentDB (mit MongoDB-Kompatibilität) bietet eine zeitlich geordnete Abfolge von Änderungsereignissen, die in den Sammlungen Ihres Clusters auftreten. Sie können Ereignisse aus einem Change Stream lesen, um zahlreiche verschiedene Anwendungsfälle zu implementieren, einschließlich:

- Änderungsbenachrichtigung
- Volltextsuche mit Amazon OpenSearch Service (OpenSearch Service)
- Analytik mit Amazon Redshift

Anwendungen können Änderungsstreams verwenden, um Datenveränderungen bei individuellen Sammlungen zu abonnieren. Die Ereignisse werden in Change Streams in der Reihenfolge angeordnet, wie sie im Cluster auftreten, und nach der Aufzeichnung des Ereignisses 3 Stunden (Standardeinstellung) gespeichert. Die Aufbewahrungsfrist kann mithilfe des `change_stream_log_retention_duration` Parameters auf bis zu 7 Tage verlängert werden. Informationen zum Ändern der Aufbewahrungsdauer für Change-Stream-Protokolle finden Sie unter Aufbewahrungsdauer für Change-Stream-Protokolle ändern.

Themen

- Unterstützte Vorgänge
- Fakturierung
- Einschränkungen

- Change-Streams aktivieren
- Beispiel: Change-Streams mit Python verwenden
- Vollständige Suche nach Dokumenten
- Wiederaufnahme eines Change-Streams
- Wiederaufnahme eines Change-Streams mit startAtOperationTime
- Wiederaufnahme eines Change-Streams mit postBatchResumeToken
- Transaktionen in Change-Streams
- Änderung der Aufbewahrungsdauer des Change-Stream-Protokolls
- Verwendung von Change-Streams auf sekundären Instanzen

Unterstützte Vorgänge

Amazon DocumentDB unterstützt die folgenden Operationen für Change-Streams:

- Alle Änderungsereignisse werden in der MongoDB `db.collection.watch()` `db.watch()` und der `client.watch()` API unterstützt.
- Vollständige Dokumentsuche nach Aktualisierungen.
- Aggregationsphasen:`$match`, `$project``$redact`, und und`$addFields`. `$replaceRoot`
- Einen Change-Stream von einem Resume-Token aus fortsetzen
- Wiederaufnahme eines Change-Streams aus einem Zeitstempel mit `startAtOperation` (gilt für Amazon DocumentDB 4.0+)

Fakturierung

Die Amazon DocumentDB DocumentDB-Funktion zum Ändern von Streams ist standardmäßig deaktiviert und es fallen keine zusätzlichen Gebühren an, bis die Funktion aktiviert ist. Die Verwendung von Change-Streams in einem Cluster verursacht zusätzliche Lese-, Schreib- IOs und Speicherkosten. Sie können den `modifyChangeStreams` API-Vorgang verwenden, um diese Funktion für Ihren Cluster zu aktivieren. Weitere Informationen zur Preisgestaltung finden Sie unter Amazon DocumentDB DocumentDB-Preise.

Einschränkungen

Für Change-Streams gelten in Amazon DocumentDB die folgenden Einschränkungen:

- Auf Amazon DocumentDB 3.6 und Amazon DocumentDB 4.0 können Change-Streams nur über eine Verbindung zur primären Instance eines Amazon DocumentDB-Clusters geöffnet werden. Das Lesen aus Change-Streams auf einer Replikat-Instance wird in Amazon DocumentDB 3.6 und Amazon DocumentDB 4.0 nicht unterstützt. Beim Aufruf der API-Operation `watch()` müssen Sie die Leseeinstellung `primary` angeben, um sicherzustellen, dass alle Lesevorgänge an die primäre Instance weitergeleitet werden (siehe den Abschnitt Beispiel).
- In Amazon DocumentDB 5.0 können Change-Streams sowohl von primären Instances als auch von sekundären Instances, einschließlich globaler Cluster, geöffnet werden. Sie können eine sekundäre Lesepräferenz angeben, um die Change-Streams an sekundäre Instances umzuleiten. Verwendung von Change-Streams auf sekundären InstanzenWeitere bewährte Methoden und Einschränkungen finden Sie unter.
- Ereignisse, die in einen Change-Stream für eine Sammlung geschrieben wurden, sind bis zu 7 Tage lang verfügbar (die Standardeinstellung ist 3 Stunden). Änderungsstream-Daten werden nach dem Zeitfenster der Protokollaufbewahrungsdauer gelöscht, auch wenn keine neuen Änderungen vorgenommen wurden.
- Eine über längere Zeit für eine Sammlung ausgeführte Schreiboperation wie `updateMany` oder `deleteMany` kann das Schreiben von Change Stream-Ereignissen vorübergehend bis zum Abschluss der über längere Zeit ausgeführten Schreiboperation blockieren.
- Amazon DocumentDB unterstützt das MongoDB-Betriebsprotokoll () `oplog` nicht.
- Bei Amazon DocumentDB müssen Sie Change-Streams für eine bestimmte Sammlung explizit aktivieren.
- Wenn die Gesamtgröße eines Change Stream-Ereignisses (einschließlich der Änderungsdaten und des vollständigen Dokuments, wenn angefordert) größer als `16 MB` ist, tritt auf dem Client ein Lesefehler für die Change Streams auf.
- Der Ruby-Treiber wird derzeit nicht unterstützt, wenn `db.watch()` und `client.watch()` mit Amazon DocumentDB 3.6 verwendet wird.
- Die Ausgabe des `updateDescription` Befehls in Change-Streams unterscheidet sich in Amazon DocumentDB von der in MongoDB, wenn der aktualisierte Wert des Felds derselbe ist wie der vorherige:
  - Amazon DocumentDB gibt kein Feld in der `updateDescription` Ausgabe zurück, wenn das angegebene Feld im `$set` Befehl angegeben ist und sein Zielwert bereits dem Quellwert entspricht.
  - MongoDB gibt das Feld in der Ausgabe zurück, auch wenn der angegebene Wert dem aktuellen Wert entspricht.

Change-Streams aktivieren

Sie können Amazon DocumentDB DocumentDB-Change-Streams für alle Sammlungen innerhalb einer bestimmten Datenbank oder nur für ausgewählte Sammlungen aktivieren. Im Folgenden finden Sie Beispiele für die Aktivierung von Change Streams für verschiedene Anwendungsfälle über die Mongo-Shell. Leere Zeichenfolgen werden bei der Angabe von Datenbank- und Sammlungsnamen als Platzhalter behandelt.

```
//Enable change streams for the collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "foo",
    enable: true});
```

```
//Disable change streams on collection "foo" in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "foo",
    enable: false});
```

```
//Enable change streams for all collections in database "bar"
db.adminCommand({modifyChangeStreams: 1,
    database: "bar",
    collection: "",
    enable: true});
```

```
//Enable change streams for all collections in all databases in a cluster
db.adminCommand({modifyChangeStreams: 1,
    database: "",
    collection: "",
    enable: true});
```

Change Streams werden für eine Sammlung aktiviert, wenn eine der folgenden Bedingungen erfüllt ist:

- Sowohl die Datenbank als auch die Sammlung sind explizit aktiviert.
- Die Datenbank, die die Sammlung enthält, ist aktiviert.
- Alle Datenbanken sind aktiviert.

Wenn Sie eine Sammlung aus einer Datenbank löschen, werden Change Streams für diese Sammlung nicht deaktiviert, wenn Change Streams für die übergeordnete Datenbank auch aktiviert sind oder wenn alle Datenbanken im Cluster aktiviert sind. Wenn eine neue Sammlung mit demselben Namen wie die gelöschte Sammlung erstellt wird, werden Change Streams für diese Sammlung aktiviert.

Sie können alle aktivierten Change Streams Ihres Clusters mithilfe der `$listChangeStreams`-Aggregationspipeline-Phase auflisten. Alle von Amazon DocumentDB unterstützten Aggregationsphasen können in der Pipeline für die weitere Verarbeitung verwendet werden. Wenn eine zuvor aktivierte Sammlung deaktiviert wurde, wird sie nicht in der `$listChangeStreams`-Ausgabe angezeigt.

```
//List all databases and collections with change streams enabled
cursor = new DBCommandCursor(db,
    db.runCommand(
        {aggregate: 1,
        pipeline: [{$listChangeStreams: 1}],
        cursor:{}}));
```

```
//List of all databases and collections with change streams enabled
{ "database" : "test", "collection" : "foo" }
{ "database" : "bar", "collection" : "" }
{ "database" : "", "collection" : "" }
```

```
//Determine if the database “bar” or collection “bar.foo” have change streams enabled
cursor = new DBCommandCursor(db,
  db.runCommand(
      {aggregate: 1,
       pipeline: [{$listChangeStreams: 1},
                  {$match: {$or: [{database: "bar", collection: "foo"},
                                  {database: "bar", collection: ""},
                                  {database: "", collection: ""}]}}
                ],
      cursor:{}}));
```

Beispiel: Change-Streams mit Python verwenden

Im Folgenden finden Sie ein Beispiel für die Verwendung eines Amazon DocumentDB DocumentDB-Change-Streams mit Python auf Sammlungsebene.

```
import os
import sys
from pymongo import MongoClient, ReadPreference

username = "DocumentDBusername"
password = <Insert your password>

clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']

#While ‘Primary’ is the default read preference, here we give an example of
#how to specify the required read preference when reading the change streams
coll = db.get_collection('foo', read_preference=ReadPreference.PRIMARY)
#Create a stream object
stream = coll.watch()
#Write a new document to the collection to generate a change event
coll.insert_one({'x': 1})
#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f600000002010000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'insert'}
"""

#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
Expected Output:
None
"""

#Generate a new change event by updating a document
```

```
result = coll.update_one({'x': 1}, {'$set': {'x': 2}})
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf99d40000000101000000100009025'},
'clusterTime': Timestamp(1571789268, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 2}}}
"""
```

Das Folgende ist ein Beispiel für die Verwendung eines Amazon DocumentDB DocumentDB-Change-Streams mit Python auf Datenbankebene.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']
#Create a stream object
stream = db.watch()
coll = db.get_collection('foo')
#Write a new document to the collection foo to generate a change event
coll.insert_one({'x': 1})

#Read the next change event from the stream (if any)
print(stream.try_next())

"""
Expected Output:
{'_id': {'_data': '015daf94f60000000201000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo', 'db': 'bar'},
```

```
'operationType': 'insert'}
"""
#A subsequent attempt to read the next change event returns nothing, as there are no
 new changes
print(stream.try_next())

"""
Expected Output:
None
"""

coll = db.get_collection('foo1')

#Write a new document to another collection to generate a change event
coll.insert_one({'x': 1})
print(stream.try_next())

"""
Expected Output: Since the change stream cursor was the database level you can see
 change events from different collections in the same database
{'_id': {'_data': '015daf94f60000000201000000200009025'},
'clusterTime': Timestamp(1571788022, 2),
'documentKey': {'_id': ObjectId('5daf94f6ea258751778163d6')},
'fullDocument': {'_id': ObjectId('5daf94f6ea258751778163d6'), 'x': 1},
'ns': {'coll': 'foo1', 'db': 'bar'},
'operationType': 'insert'}
"""
```

Vollständige Suche nach Dokumenten

Das Änderungsereignis „Aktualisierung“ enthält nicht das vollständige Dokument, sondern lediglich die ausgeführte Änderung. Wenn für Ihren Anwendungsfall das vollständige Dokument erforderlich ist, das von einer Aktualisierung betroffen ist, können Sie beim Öffnen des Datenstroms die vollständige Dokumentsuche aktivieren.

Das `fullDocument`-Dokument für ein Update-Change Stream-Ereignis stellt die neueste Version des aktualisierten Dokuments zum Zeitpunkt der Dokumentsuche dar. Wenn zwischen der Aktualisierungsoperation und der `fullDocument`-Suche Änderungen ausgeführt wurden, besitzt das `fullDocument`-Dokument möglicherweise nicht den Dokumentstatus zur Aktualisierungszeit.

Verwenden Sie dieses Beispiel, um ein Stream-Objekt mit aktivierter Aktualisierungssuche zu erstellen:

```
stream = coll.watch(full_document='updateLookup')

#Generate a new change event by updating a document
result = coll.update_one({'x': 2}, {'$set': {'x': 3}})

stream.try_next()
```

Die Ausgabe des Stream-Objekts sieht ungefähr so aus:

```
{'_id': {'_data': '015daf9b7c000000010100000100009025'},
'clusterTime': Timestamp(1571789692, 1),
'documentKey': {'_id': ObjectId('5daf9502ea258751778163d7')},
'fullDocument': {'_id': ObjectId('5daf9502ea258751778163d7'), 'x': 3},
'ns': {'coll': 'foo', 'db': 'bar'},
'operationType': 'update',
'updateDescription': {'removedFields': [], 'updatedFields': {'x': 3}}}
```

Wiederaufnahme eines Change-Streams

Sie können einen Change Stream zu einem späteren Zeitpunkt mithilfe eines Fortsetzungs-Tokens fortsetzen, das dem Feld `_id` des zuletzt abgerufenen Änderungsereignisdokuments entspricht.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem', retryWrites='false')

db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
token = event['_id']
print(token)

"""
```

```
Output: This is the resume token that we will later us to resume the change stream
{'_data': '015daf9c5b0000000101000000100009025'}
"""
#Python provides a nice shortcut for getting a stream's resume token
print(stream.resume_token)

"""
Output
{'_data': '015daf9c5b0000000101000000100009025'}
"""
#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
#Generate another change event by inserting a document
result = coll.insert_one({'y': 5})
#Open a stream starting after the selected resume token
stream = db.watch(full_document='updateLookup', resume_after=token)
#Our first change event is the update with the specified _id
print(stream.try_next())

"""
#Output: Since we are resuming the change stream from the resume token, we will see all
 events after the first update operation. In our case, the change stream will resume
 from the update operation {x:5}

{'_id': {'_data': '015f7e8f0c0000000601000000006000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602129676, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0ac423bafbfd9adba2')},
'fullDocument': {'_id': ObjectId('5f7e8f0ac423bafbfd9adba2'), 'x': 5},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""
#Followed by the insert
print(stream.try_next())

"""
#Output:
{'_id': {'_data': '015f7e8f0c0000000701000000007000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602129676, 7),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94')},
'fullDocument': {'_id': ObjectId('5f7e8f0cbf8c233ed577eb94'), 'y': 5}}
```

```
"""
```

Wiederaufnahme eines Change-Streams mit **`startAtOperationTime`**

Sie können einen Änderungsstream zu einem späteren Zeitpunkt ab einem bestimmten Zeitstempel wieder aufnehmen, indem Sie`startAtOperationTime`.

> **Note**
>
> Die Fähigkeit zur Verwendung `startAtOperationTime` ist in Amazon DocumentDB 4.0+ verfügbar. Bei Verwendung `startAtOperationTime` gibt der Change-Stream-Cursor nur Änderungen zurück, die zu oder nach dem angegebenen Zeitstempel vorgenommen wurden. Die `resumeAfter` Befehle `startAtOperationTime` und schließen sich gegenseitig aus und können daher nicht zusammen verwendet werden.

```
import os
import sys
from pymongo import MongoClient

username = "DocumentDBusername"
password = <Insert your password>
clusterendpoint = "DocumentDBClusterEndpoint”
client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='rds-root-ca-2020.pem',retryWrites='false')
db = client['bar']
coll = db.get_collection('foo')
#Create a stream object
stream = db.watch()
coll.update_one({'x': 1}, {'$set': {'x': 4}})
event = stream.try_next()
timestamp = event['clusterTime']
print(timestamp)
"""
Output
Timestamp(1602129114, 4)
"""
#Generate a new change event by updating a document
result = coll.update_one({'x': 4}, {'$set': {'x': 5}})
result = coll.insert_one({'y': 5})
#Generate another change event by inserting a document
```

```
#Open a stream starting after specified time stamp

stream = db.watch(start_at_operation_time=timestamp)
print(stream.try_next())

"""
#Output: Since we are resuming the change stream at the time stamp of our first update
 operation (x:4), the change stream cursor will point to that event
{'_id': {'_data': '015f7e941a0000000301000000003000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602130970, 3),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e9417c423bafbfd9adbb1')},
'updateDescription': {'updatedFields': {'x': 4}, 'removedFields': []}}
"""

print(stream.try_next())
"""
#Output: The second event will be the subsequent update operation (x:5)
{'_id': {'_data': '015f7e9502000000050100000005000fe038'},
'operationType': 'update',
'clusterTime': Timestamp(1602131202, 5),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e94ffc423bafbfd9adbb2')},
'updateDescription': {'updatedFields': {'x': 5}, 'removedFields': []}}
"""

print(stream.try_next())

"""
#Output: And finally the last event will be the insert operation (y:5)
{'_id': {'_data': '015f7e9502000000060100000006000fe038'},
'operationType': 'insert',
'clusterTime': Timestamp(1602131202, 6),
'ns': {'db': 'bar', 'coll': 'foo'},
'documentKey': {'_id': ObjectId('5f7e95025c4a569e0f6dde92')},
'fullDocument': {'_id': ObjectId('5f7e95025c4a569e0f6dde92'), 'y': 5}}
"""
```

Wiederaufnahme eines Change-Streams mit **postBatchResumeToken**

Der Amazon DocumentDB DocumentDB-Change-Stream gibt jetzt ein zusätzliches Feld namens `postBatchResumeToken` zurück. Dieses Feld wird vom `$changestream` Befehl und `getMore` vom Befehl zurückgegeben.

Beispiel für den `$changestream` Befehl in Python:

```
db.command({"aggregate": "sales", "pipeline": [{ "$changeStream": {}}], "cursor":
 {"batchSize": 1}})
```

Erwartete Ausgabe:

```
cursor" : {
   "firstBatch" : [ ],
   "postBatchResumeToken" : {"_data" : "0167c8cbe60000000004"},
   "id" : NumberLong("9660788144470"),
   "ns" : "test.sales"
}
```

Beispiel für den `getMore` Befehl in Python:

```
db.command({"getMore": NumberLong(<cursor id>), "collection": "sales", "batchSize":
 1 })
```

Erwartete Ausgabe

```
cursor" : {
   "nextBatch" : [ ],
   "postBatchResumeToken" : {"_data" : "0167c8cbe60000000004"},
   "id" : NumberLong("9660788144470"),
   "ns" : "test.sales"
}
```

Das `postBatchResumeToken` Feld kann verwendet werden, um neue Change-Stream-Cursor im `resumeAfter` Feld zu öffnen, ähnlich wie das Resume-Token verwendet wird.

Öffnet einen Stream, der nach dem ausgewählten `postBatchResumeToken` beginnt:

```
post_batch_resume_token = output['cursor']['postBatchResumeToken']
```

```
stream = db.watch(full_document='updateLookup', resume_after=post_batch_resume_token)
```

Im Gegensatz zu einem regulären Resume-Token, das immer einem Operationsprotokolleintrag (Oplog) entspricht, der ein aktuelles Ereignis widerspiegelt, `postBatchResumeToken` entspricht dies einem Oplog-Eintrag, bis zu dem der Change-Stream auf dem Server gescannt hat. Dabei handelt es sich nicht unbedingt um eine entsprechende Änderung.

Der Versuch, den Vorgang mit einem alten regulären Resume-Token fortzusetzen, zwingt die Datenbank, alle Oplog-Einträge zwischen dem angegebenen Zeitstempel und der aktuellen Uhrzeit zu scannen. Dadurch können intern viele Abfragen generiert werden, wobei jede Unterabfrage für einen kurzen Zeitraum gescannt wird. Dies führt zu einem Anstieg der CPU-Auslastung und beeinträchtigt die Datenbankleistung. Wenn Sie mit dem letzten `postBatchResumeToken` Schritt fortfahren, wird das Scannen nicht übereinstimmender Protokolleinträge übersprungen.

Transaktionen in Change-Streams

Change-Stream-Ereignisse enthalten keine Ereignisse aus nicht bestätigten, and/or abgebrochenen Transaktionen. Wenn Sie beispielsweise eine Transaktion mit einem Vorgang und einem `INSERT` Vorgang starten und Ihr `UPDATE` Vorgang erfolgreich ist, der `INSERT` `UPDATE` Vorgang jedoch fehlschlägt, wird die Transaktion zurückgesetzt. Da diese Transaktion zurückgesetzt wurde, enthält Ihr Change-Stream keine Ereignisse für diese Transaktion.

Änderung der Aufbewahrungsdauer des Change-Stream-Protokolls

Sie können die Aufbewahrungsdauer für das Change-Stream-Protokoll mit AWS Management Console oder auf einen Wert zwischen 1 Stunde und 7 Tagen ändern AWS CLI.

Using the AWS Management Console

So ändern Sie den Aufbewahrungszeitraum für das Change-Stream-Protokoll

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol

(≡)
aus.

3. Wählen Sie im Bereich Parameter groups (Parametergruppen) die Cluster-Parametergruppe, die Ihrem Cluster zugeordnet ist. Informationen zum Identifizieren der Clusterparametergruppe, die dem Cluster zugeordnet ist, finden Sie unter Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters.
4. Auf der resultierenden Seite werden die Parameter und die entsprechenden Details zu Ihrer Cluster-Parametergruppe angezeigt. Wählen Sie den `change_stream_log_retention_duration`-Parameter aus.
5. Klicken Sie oben rechts auf der Seite auf Edit (Bearbeiten), um den Wert des Parameters zu ändern. Der `change_stream_log_retention_duration` Parameter kann so geändert werden, dass er zwischen 1 Stunde und 7 Tagen liegt.
6. Führen Sie die Änderung aus. Wählen Sie anschließend Modify cluster parameter (Cluster-Parameter ändern) aus, um die Änderungen zu speichern. Um die Änderungen zu verwerfen, wählen Sie Cancel (Abbrechen) aus.

Using the AWS CLI

Um den `change_stream_log_retention_duration`-Parameter Ihrer Cluster-Parametergruppe zu ändern, verwenden Sie die `modify-db-cluster-parameter-group`-Operation mit den folgenden Parametern:

- **`--db-cluster-parameter-group-name`** – Erforderlich. Der Name der Cluster-Parametergruppe, die Sie ändern. Informationen zum Identifizieren der Clusterparametergruppe, die dem Cluster zugeordnet ist, finden Sie unter Ermitteln der Parametergruppe eines Amazon DocumentDB-Clusters.
- **`--parameters`** – Erforderlich. Der Parameter, der von Ihnen geändert wird. Jeder Parametereintrag muss Folgendes enthalten:
  - **`ParameterName`**— Der Name des Parameters, den Sie ändern. In diesem Fall ist es `change_stream_log_retention_duration`
  - **`ParameterValue`**— Der neue Wert für diesen Parameter.
  - **`ApplyMethod`**— Wie die Änderungen an diesem Parameter angewendet werden sollen. Zugelassene Werte sind `immediate` und `pending-reboot`.

> **Note**
>
> Parameter mit dem `ApplyType` von `static` müssen über einen `ApplyMethod` von `pending-reboot` verfügen.

1. Um die Werte des Parameters `change_stream_log_retention_duration` zu ändern, führen Sie den folgenden Befehl aus und ersetzen `parameter-value` durch den Wert, auf den Sie den Parameter ändern möchten.

   Für Linux, macOS oder Unix:

   ```
   aws docdb modify-db-cluster-parameter-group \
       --db-cluster-parameter-group-name sample-parameter-group \
       --parameters
    "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-
   value>,ApplyMethod=immediate"
   ```

   Für Windows:

   ```
   aws docdb modify-db-cluster-parameter-group ^
       --db-cluster-parameter-group-name sample-parameter-group ^
       --parameters
    "ParameterName=change_stream_log_retention_duration,ParameterValue=<parameter-
   value>,ApplyMethod=immediate"
   ```

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

   ```
   {
       "DBClusterParameterGroupName": "sample-parameter-group"
   }
   ```

2. Warten Sie mindestens 5 Minuten.
3. Listen Sie die Parameterwerte von `sample-parameter-group` auf, um sicherzustellen, dass Ihre Änderungen übernommen wurden.

   Für Linux, macOS oder Unix:

   ```
   aws docdb describe-db-cluster-parameters \
   ```

```
    --db-cluster-parameter-group-name sample-parameter-group
```

Für Windows:

```
aws docdb describe-db-cluster-parameters ^
    --db-cluster-parameter-group-name sample-parameter-group
```

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

```
{
    "Parameters": [
        {
            "ParameterName": "audit_logs",
            "ParameterValue": "disabled",
            "Description": "Enables auditing on cluster.",
            "Source": "system",
            "ApplyType": "dynamic",
            "DataType": "string",
            "AllowedValues": "enabled,disabled",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "change_stream_log_retention_duration",
            "ParameterValue": "12345",
            "Description": "Duration of time in seconds that the change stream
 log is retained and can be consumed.",
            "Source": "user",
            "ApplyType": "dynamic",
            "DataType": "integer",
            "AllowedValues": "3600-86400",
            "IsModifiable": true,
            "ApplyMethod": "immediate"
        }
    ]
}
```

> **Note**
>
> Bei der Aufbewahrung von Change-Stream-Protokollen werden Protokolle, die älter als der konfigurierte `change_stream_log_retention_duration` Wert sind, erst gelöscht, wenn die Protokollgröße größer als (>) 51.200 MB ist.

Verwendung von Change-Streams auf sekundären Instanzen

Um mit der Verwendung von Change Stream auf sekundären Instances zu beginnen, öffnen Sie den Change-Stream-Cursor `readPreference` als sekundäre Instanz.

Sie können einen Change-Stream-Cursor öffnen, um nach Änderungsereignissen in einer bestimmten Sammlung oder allen Sammlungen in einem Cluster oder einer Datenbank Ausschau zu halten. Sie können einen Change-Stream-Cursor auf jeder Amazon DocumentDB DocumentDB-Instance öffnen und Change-Stream-Dokumente sowohl von Writer- als auch von Reader-Instances abrufen. Sie können Change-Stream-Tokens (wie `resumeToken` oder`startOperationTime`) für verschiedene Change-Stream-Cursors, die auf einer Writer- und Reader-Instance geöffnet sind, gemeinsam nutzen.

Beispiel

```
import os
import sys
from pymongo import MongoClient, ReadPreference

username = "DocumentDBusername"
password = <Your password>

clusterendpoint = "DocumentDBClusterEndpoint"

client = MongoClient(clusterendpoint, username=username, password=password, tls='true',
 tlsCAFile='global-bundle.pem')

db = client['bar']

# Make sure to use SECONDARY to redirect cursor reads from secondary instances
coll = db.get_collection('foo', read_preference=ReadPreference.SECONDARY)

# Create a stream object on RO. The token needs to generated from PRIMARY.
stream = coll.watch(resumeAfter=token)
```

```
for event in stream:
   print(event)
```

Richtlinien und Einschränkungen für Change-Streams auf sekundären Instanzen

- Change-Stream-Ereignisse müssen von der primären Instance auf die sekundären Instances repliziert werden. Sie können die Verzögerung anhand der `DBInstanceReplicaLag` Metrik in Amazon überwachen CloudWatch.
- Zeitstempel auf sekundären Instances stimmen möglicherweise nicht immer mit denen der primären Instance überein. In diesem Fall müssen Sie mit Verzögerungen beim Zeitstempel der sekundären Instanz rechnen, damit diese aufholen kann. Als bewährte Methode empfehlen wir, die Überwachung auf der sekundären Instanz `resumeToken` zu verwenden `startAtOperationTime` oder zu starten.
- Auf sekundären Instanzen kann es zu einem geringeren Durchsatz kommen als auf der primären Instanz, wenn Ihr Dokument groß ist, Sie dies tun`fullDocumentLookup`, und auf der primären Instanz eine hohe Arbeitslast beim gleichzeitigen Schreiben anfällt. Als bewährte Methode empfehlen wir, die Trefferquote im Puffercache auf der sekundären Ebene zu überwachen und sicherzustellen, dass die Trefferquote im Puffercache hoch ist.

Verwendung AWS Lambda mit Change-Streams

Amazon DocumentDB ist integriert, AWS Lambda sodass Sie Lambda-Funktionen verwenden können, um Datensätze in einem Change-Stream zu verarbeiten. Die Zuordnung von Lambda-Ereignisquellen ist eine Ressource, mit der Lambda-Funktionen aufgerufen werden können, um Amazon DocumentDB DocumentDB-Ereignisse zu verarbeiten, die Lambda nicht direkt aufrufen. Mit Amazon DocumentDB Change Stream als Ereignisquelle können Sie ereignisgesteuerte Anwendungen erstellen, die auf Änderungen in Ihren Daten reagieren. Sie können Lambda-Funktionen beispielsweise verwenden, um neue Dokumente zu verarbeiten, Aktualisierungen vorhandener Dokumente nachzuverfolgen oder gelöschte Dokumente zu protokollieren.

Sie können eine Ereignisquellenzuordnung so konfigurieren, dass Datensätze aus Ihrem Amazon DocumentDB DocumentDB-Change-Stream an eine Lambda-Funktion gesendet werden. Ereignisse können einzeln oder gebündelt gesendet werden, um die Effizienz zu verbessern. Sie werden der Reihe nach verarbeitet. Sie können das Batchverhalten Ihrer Ereignisquellenzuordnung auf der Grundlage einer bestimmten Zeitfensterdauer (0-300 Sekunden) oder der Anzahl von Batch-Datensätzen (maximale Grenze von 10.000 Datensätzen) konfigurieren. Sie können mehrere

Ereignisquellenzuordnungen erstellen, um dieselben Daten mit mehreren Lambda-Funktionen zu verarbeiten oder um unterschiedliche Elemente aus mehreren Streams mit einer einzigen Funktion zu verarbeiten.

Wenn Ihre Funktion einen Fehler zurückgibt, versucht Lambda den Batch erneut, bis er erfolgreich verarbeitet wurde. Falls die Ereignisse im Change-Stream abgelaufen sind, deaktiviert Lambda die Zuordnung der Ereignisquellen. In diesem Fall können Sie eine neue Zuordnung der Ereignisquelle erstellen und diese mit einer Startposition Ihrer Wahl konfigurieren. Lambda-Ereignisquellenzuordnungen verarbeiten Ereignisse aufgrund der verteilten Natur ihrer Poller mindestens einmal. Infolgedessen kann Ihre Lambda-Funktion in seltenen Situationen doppelte Ereignisse erhalten. Folgen Sie den bewährten Methoden für die Arbeit mit AWS Lambda Funktionen und erstellen Sie idempotente Funktionen, um Probleme im Zusammenhang mit doppelten Ereignissen zu vermeiden. Weitere Informationen finden Sie unter Using AWS Lambda console with Amazon DocumentDB im AWS Lambda Developer Guide.

Gemäß bewährten Methoden für die Leistung muss die Lambda-Funktion kurzlebig sein. Um unnötige Verarbeitungsverzögerungen zu vermeiden, sollte sie auch keine komplexe Logik ausführen. Insbesondere bei einem Hochgeschwindigkeits-Stream ist es besser, asynchrone Nachbearbeitungs-Schrittfunktions-Workflows auszulösen als synchrone Lambdas mit langer Laufzeit. Weitere Informationen zu AWS Lambda finden Sie im AWS Lambda Entwicklerhandbuch.

Einschränkungen

Die folgenden Einschränkungen sind bei der Arbeit mit Amazon DocumentDB zu beachten und AWS Lambda:

- AWS Lambda wird derzeit nur auf Amazon DocumentDB 4.0 und 5.0 unterstützt.
- AWS Lambda wird derzeit nicht auf elastischen Clustern oder globalen Clustern unterstützt.
- AWS Lambda Die Größe der Nutzlast darf 6 MB nicht überschreiten. Weitere Informationen zu Lambda-Batchgrößen finden Sie unter „Batchverhalten“ im Abschnitt Lambda-Ereignisquellenzuordnungen im Entwicklerhandbuch.AWS Lambda

Verwendung elastischer Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster unterstützen Workloads mit einer reads/writes Speicherkapazität von Millionen pro Sekunde und Petabyte. Elastische Cluster vereinfachen auch die Interaktion von Entwicklern mit Amazon DocumentDB, da sie Instances nicht mehr auswählen, verwalten oder aktualisieren müssen.

Elastische Amazon DocumentDB-Cluster wurden erstellt, um:

- Bieten Sie eine Lösung für Kunden, die nach einer Datenbank suchen, die praktisch grenzenlose Skalierbarkeit mit umfangreichen Abfragefunktionen und MongoDB-API-Kompatibilität bietet.
- Bieten Sie Kunden höhere Verbindungslimits und reduzieren Sie Ausfallzeiten aufgrund von Patches.
- Investieren Sie weiterhin in eine cloudnative, elastische und erstklassige Architektur für JSON-Workloads.

Themen

- Anwendungsfälle für elastische Cluster
- Vorteile elastischer Cluster
- Elastische Cluster-Region und Versionsverfügbarkeit
- Einschränkungen
- Elastische Amazon DocumentDB-Cluster: So funktioniert's
- Erste Schritte mit Amazon DocumentDB Elastic Clusters
- Bewährte Methoden für Amazon DocumentDB Elastic Cluster
- Verwaltung elastischer Amazon DocumentDB-Cluster
- Datenverschlüsselung im Ruhezustand für elastische Amazon DocumentDB-Cluster
- Serviceverknüpfte Rollen in elastischen Clustern

Anwendungsfälle für elastische Cluster

Dokumentdatenbanken sind nützlich für Workloads, die ein flexibles Schema für eine schnelle, iterative Entwicklung benötigen. Beispiele für Amazon DocumentDB DocumentDB-Anwendungsfälle finden Sie unterAnwendungsfälle für Dokumentendatenbanken.

Im Folgenden finden Sie einige Beispiele für Anwendungsfälle, für die elastische Cluster erhebliche Vorteile bieten können:

Benutzerprofile

Da Dokumentendatenbanken über ein flexibles Schema verfügen, können sie Dokumente mit unterschiedlichen Attributen und Datenwerten skaliert speichern. Elastische Cluster sind eine praktische Lösung für Online-Profile, in denen verschiedene Benutzer unterschiedliche Arten von Informationen bereitstellen. Angenommen, Ihre Anwendungen unterstützen Hunderte Millionen von Benutzerprofilen. Sie können elastische Cluster zur Unterstützung solcher Anwendungen verwenden, da sie hoch- und skaliert werden können, um Millionen von Schreib- und Lesevorgängen in diese Benutzerprofile zu unterstützen. Sie können auch außerhalb der Spitzenzeiten herunterskalieren, um die Kosten zu senken.

Inhaltsverwaltung und historische Aufzeichnungen

Um Inhalte effektiv zu verwalten, müssen Sie in der Lage sein, Inhalte aus einer Vielzahl von Quellen zu sammeln, zu aggregieren und dann an den Client zu liefern. Aufgrund ihres flexiblen Schemas sind Dokumentdatenbanken ideal für die Erfassung und Speicherung jeglicher Art von Daten. Sie können sie verwenden, um neue Arten von Inhalten zu erstellen und zu integrieren, einschließlich benutzergenerierter Inhalte wie Bilder, Kommentare und Videos. Im Laufe der Zeit benötigt Ihre Datenbank möglicherweise mehr Speicherplatz. Mit elastischen Clustern können Sie Ihre Daten auf mehr Speichervolumen verteilen, sodass Sie Petabyte an Daten in einem einzigen Cluster speichern können.

Vorteile elastischer Cluster

AWS Serviceintegration

Elastische Amazon DocumentDB-Cluster lassen sich auf dieselbe Weise in andere AWS Services integrieren wie Amazon DocumentDB:

- Migration — Sie können AWS Database Migration Service (DMS) verwenden, um von MongoDB und anderen relationalen Datenbanken zu elastischen Amazon DocumentDB-Clustern zu migrieren.
- Überwachung — Sie können den Zustand und die Leistung Ihres Elastic Clusters mithilfe von Amazon überwachen CloudWatch.

- Sicherheit — Sie können Authentifizierung und Autorisierung über AWS Identity and Access Management (IAM) einrichten, um Ihre Elastic Cluster-Ressourcen wie die Erstellung und Aktualisierung eines Elastic Clusters zu verwalten und Amazon VPC für sichere reine VPC-Verbindungen verwenden. Die Anmeldung mit IAM-Authentifizierung für Elastic Clusters wird nicht unterstützt.
- Datenmanagement — Sie können Daten AWS Glue mit from/to anderen AWS Diensten wie Amazon S3, Amazon Redshift und Amazon OpenSearch Service importieren und exportieren.

Elastische Cluster-Region und Versionsverfügbarkeit

Regionale Verfügbarkeit für elastische Cluster

Die folgende Tabelle zeigt die AWS Regionen, in denen Amazon DocumentDB Elastic Cluster derzeit verfügbar sind, sowie den Endpunkt für jede Region.

Name der Region	Region	Verfügbarkeitszonen
USA Ost (Nord-Virginia)	`us-east-1`	5
USA Ost (Ohio)	`us-east-2`	3
USA West (Oregon)	`us-west-2`	3
Asien-Pazifik (Hongkong)	`ap-east-1`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	3
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Kanada (Zentral)	`ca-central-1`	3
Südamerika (São Paulo)	`sa-east-1`	3

Name der Region	Region	Verfügbarkeitszonen
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3

Verfügbarkeit von Versionen

Elastic Cluster unterstützen das MongoDB 5.0-kompatible Wire-Protokoll. Unterschiede zwischen instanzbasierten Amazon DocumentDB 4.0-Clustern und elastischen Clustern finden Sie unter. Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und Elastic Clusters

Einschränkungen

Elastische Clusterverwaltung

Die folgenden Funktionen und Funktionen zur Clusterverwaltung werden in dieser Version nicht unterstützt:

- Möglichkeit, IAM-Benutzer und -Rollen zur Authentifizierung in der Datenbank zu verwenden
- Fähigkeit, globale Cluster zu erstellen
- Bestehende Amazon DocumentDB DocumentDB-Ereignisse und Abonnieren von Veranstaltungen
- Bereichs-Sharding
- Bestehende Sammlung teilen
- Shard-Schlüssel mit mehreren Feldern
- Shard-Schlüssel ändern
- Point-in-time wiederherstellen
- Klonen
- Performance Insights

> **Note**
>
> Hinweise zu den Grenzwerten für elastische Cluster finden Sie unterAmazon DocumentDB Kontingente und Beschränkungen.

Abfrage- und Schreiboperationen

Die folgenden Befehle und Funktionen für Abfrage- und Schreiboperationen werden in dieser Version nicht unterstützt:

- DDL-Befehle bei Skalierungsvorgängen
- Profiler
- Parametergruppen
- AWS Config
- AWS Backup

Sammlungs- und Indexverwaltung

Die folgenden Funktionen zur Erfassung und Indexverwaltung werden in dieser Version nicht unterstützt:

- Einzigartige Indizes
- Nicht vollständige Indizes
- Textindizes
- Vektor-Indizes
- Komprimierung von Dokumenten

Verwaltung und Diagnose

Die folgenden Befehle und Funktionen für Verwaltung und Diagnose werden in dieser Version nicht unterstützt:

- AWS Secrets Manager
- Role-based-access-control (RBAC) benutzerdefinierte Rollen.

- Beim Herstellen einer Verbindung wird das Schreibproblem 0 nicht unterstützt.
- Ändern von Subnetzen, die zu einer VPC gehören, die derzeit keinem vorhandenen Elastic Cluster zugewiesen ist.

Opt-in-Funktionen

Die folgenden Amazon DocumentDB DocumentDB-Opt-In-Funktionen werden in dieser Version nicht unterstützt:

- ACID-Transaktionen
- DDL/DML-Prüfung
- Change streams
- Sitzungsbefehle

Elastische Amazon DocumentDB-Cluster: So funktioniert's

Die Themen in diesem Abschnitt enthalten Informationen zu den Mechanismen und Funktionen, die Amazon DocumentDB Elastic Clusters zugrunde liegen.

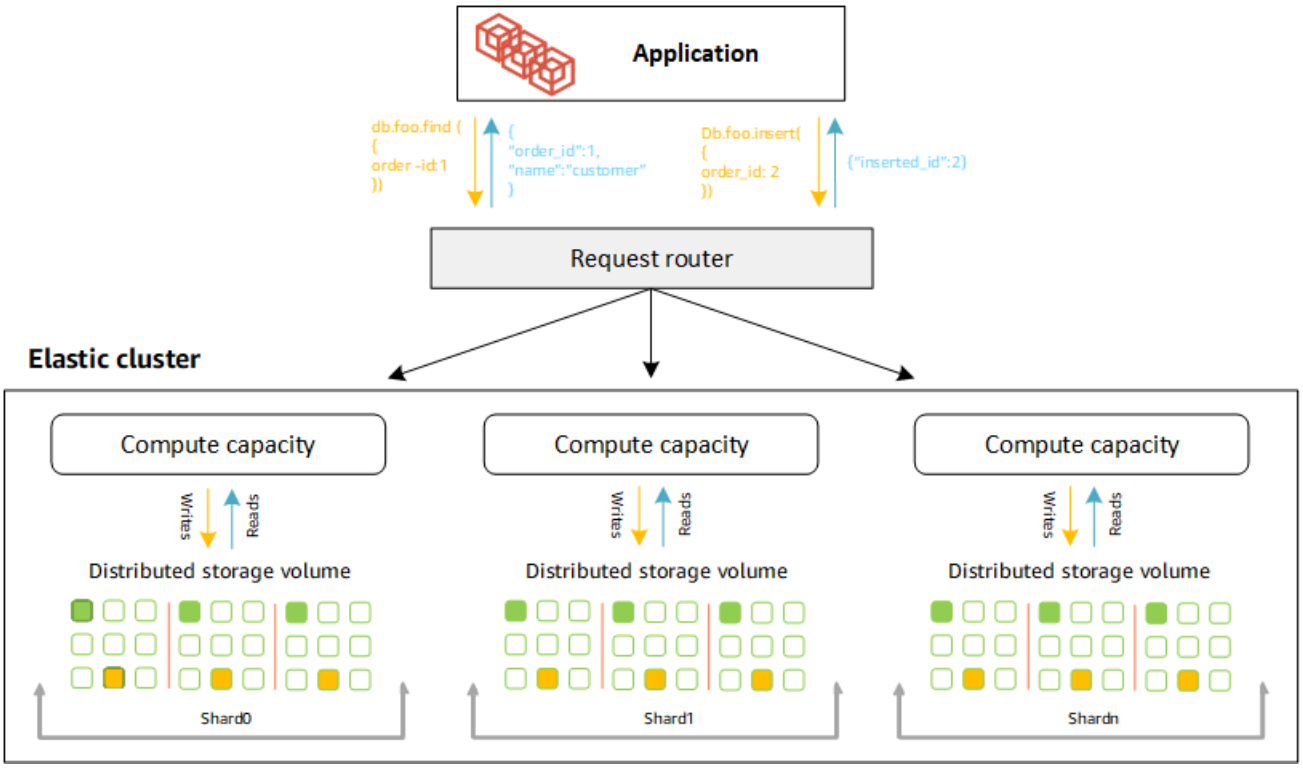
Themen

- Elastisches Cluster-Sharding von Amazon DocumentDB
- Elastische Cluster-Migration
- Elastische Cluster-Skalierung
- Elastische Cluster-Zuverlässigkeit
- Elastischer Cluster-Speicher und Verfügbarkeit
- Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und Elastic Clusters

Elastisches Cluster-Sharding von Amazon DocumentDB

Elastische Amazon DocumentDB-Cluster verwenden Hash-basiertes Sharding, um Daten in einem verteilten Speichersystem zu partitionieren. Sharding, auch Partitionierung genannt, teilt große Datensätze auf mehrere Knoten in kleine Datensätze auf, sodass Sie Ihre Datenbank über vertikale Skalierungsgrenzen hinaus skalieren können. Elastische Cluster nutzen die Trennung oder „Entkopplung“ von Rechenleistung und Speicher in Amazon DocumentDB, sodass Sie unabhängig

voneinander skalieren können. Anstatt Sammlungen neu zu partitionieren, indem kleine Datenblöcke zwischen Rechenknoten verschoben werden, kopieren elastische Cluster Daten effizient innerhalb des verteilten Speichersystems.



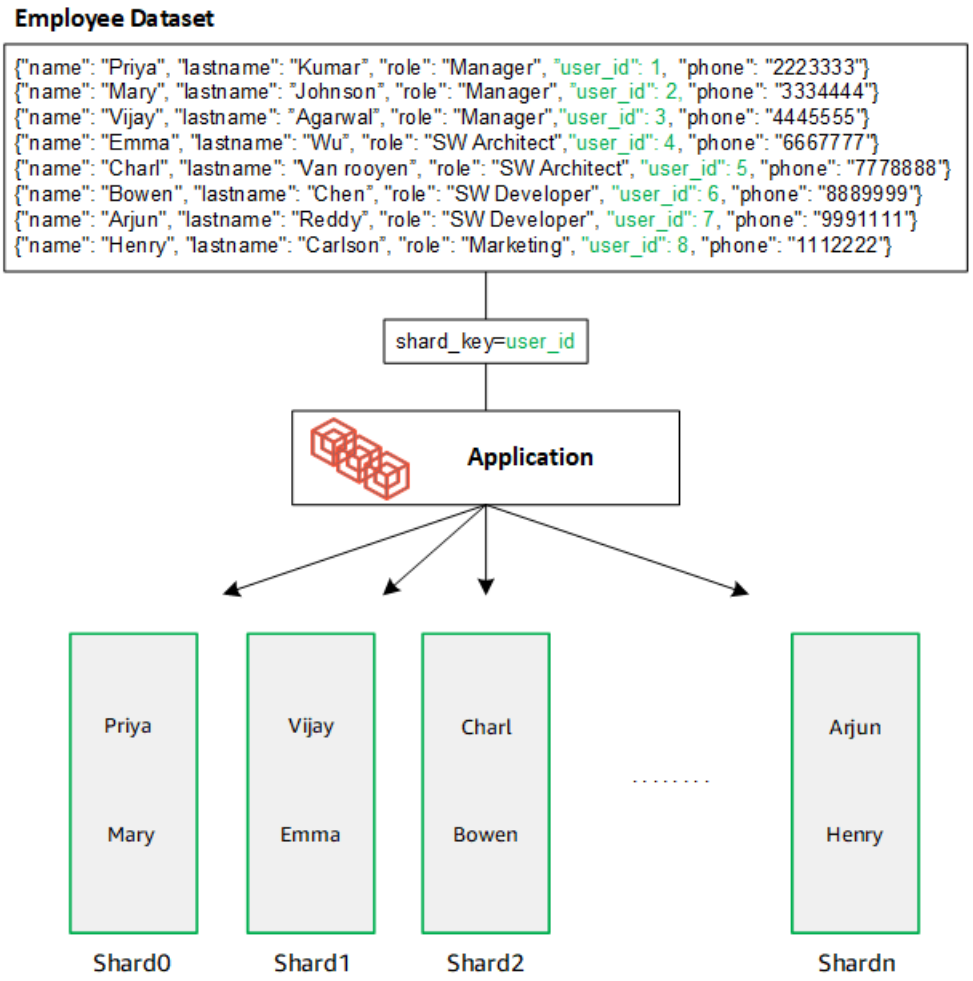
Shard-Definitionen

Definitionen der Shard-Nomenklatur:

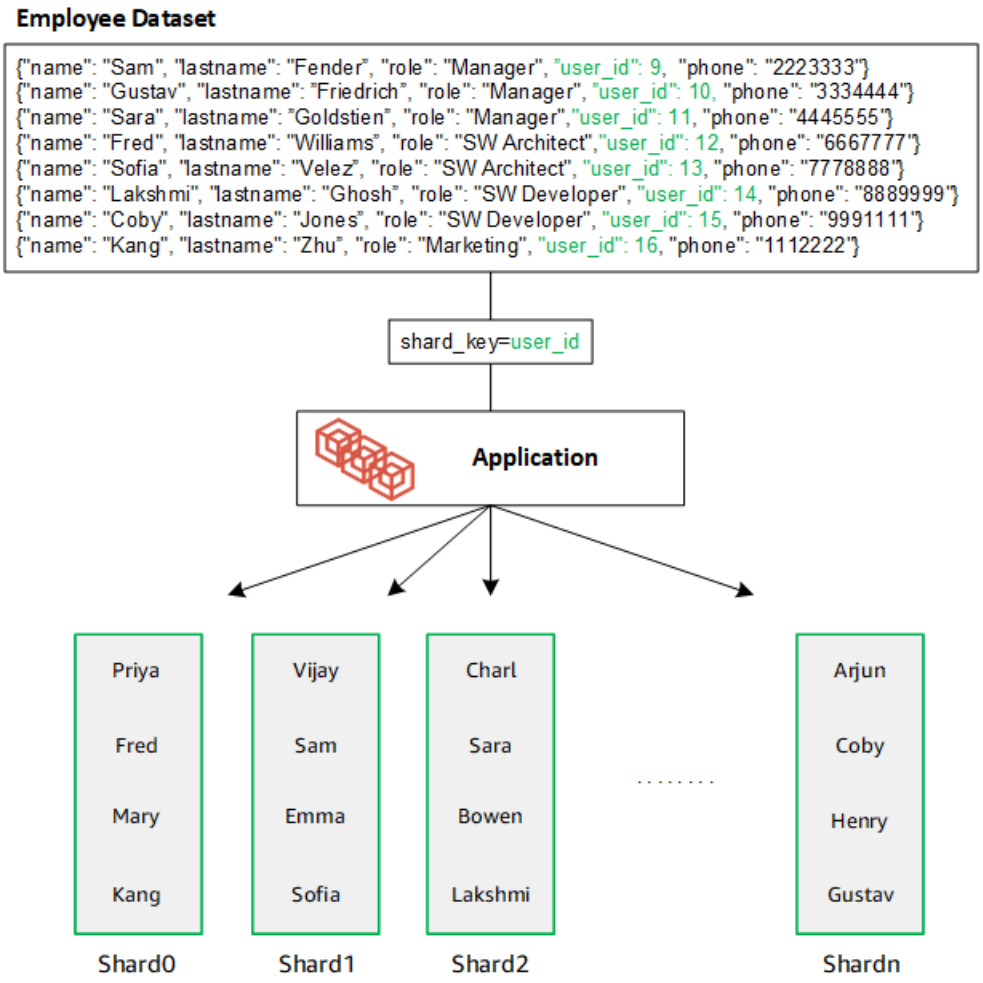
- Shard — Ein Shard stellt Rechenleistung für einen elastischen Cluster bereit. Er wird über eine einzelne Writer-Instance und 0—15 Read Replicas verfügen. Standardmäßig hat ein Shard zwei Instanzen: eine Writer- und eine einzelne Read-Replica. Sie können maximal 32 Shards konfigurieren und jede Shard-Instanz kann maximal 64 v haben. CPUs
- Shard-Schlüssel — Ein Shard-Schlüssel ist ein Pflichtfeld in Ihren JSON-Dokumenten in Sharded-Sammlungen, die Elastic Cluster verwenden, um den Lese- und Schreibverkehr auf den entsprechenden Shard zu verteilen.
- Sharded Collection — Eine Sharded Collection ist eine Sammlung, deren Daten über einen Elastic Cluster in Datenpartitionen verteilt sind.
- Partition — Eine Partition ist ein logischer Teil von Shard-Daten. Wenn Sie eine fragmentierte Sammlung erstellen, werden die Daten innerhalb jedes Shards automatisch auf der Grundlage des Shard-Schlüssels in Partitionen organisiert. Jeder Shard hat mehrere Partitionen.

Verteilung von Daten auf konfigurierte Shards

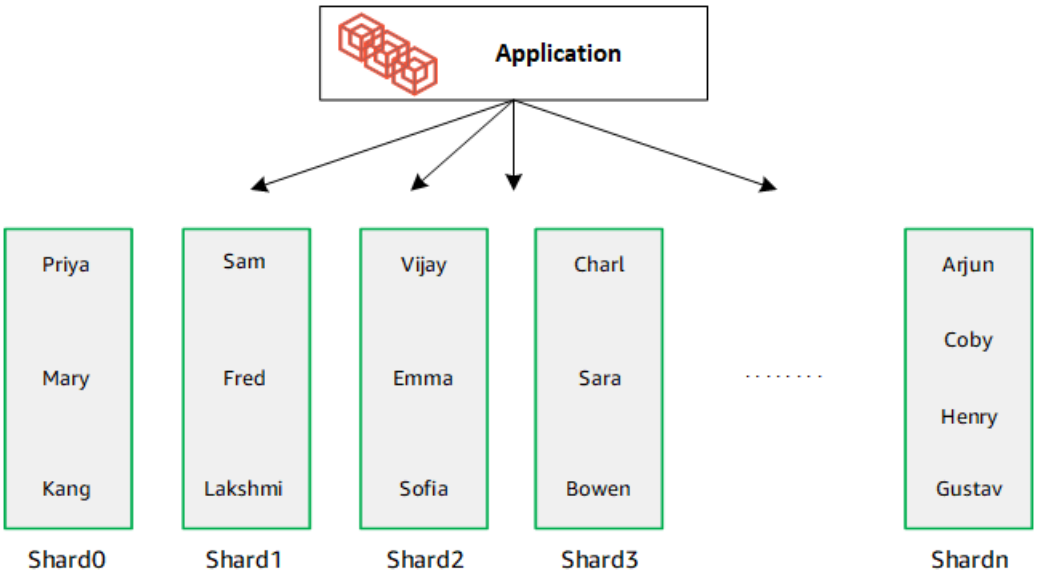
Erstellen Sie einen Shard-Schlüssel mit vielen eindeutigen Werten. Ein guter Shard-Schlüssel verteilt Ihre Daten gleichmäßig auf die zugrunde liegenden Shards, sodass Ihr Workload den besten Durchsatz und die beste Leistung erhält. Das folgende Beispiel zeigt Daten zu Mitarbeiternamen, die einen Shard-Schlüssel mit dem Namen „user_id“ verwenden:



DocumentDB verwendet Hash-Sharding, um Ihre Daten auf die zugrunde liegenden Shards zu partitionieren. Zusätzliche Daten werden auf die gleiche Weise eingefügt und verteilt:



Wenn Sie Ihre Datenbank skalieren, indem Sie zusätzliche Shards hinzufügen, verteilt Amazon DocumentDB die Daten automatisch neu:



Elastische Cluster-Migration

Amazon DocumentDB unterstützt die Migration von mongoDB-Sharded-Daten zu elastischen Clustern. Offline-, Online- und Hybridmigrationsmethoden werden unterstützt. Weitere Informationen finden Sie unter Migration zu Amazon DocumentDB.

Elastische Cluster-Skalierung

Elastische Amazon DocumentDB-Cluster bieten die Möglichkeit, die Anzahl der Shards (Scale Out) in Ihrem Elastic Cluster und die Anzahl der V, die auf jeden Shard CPUs angewendet werden, zu erhöhen (Scale Up). Sie können auch die Anzahl der Shards und die Rechenkapazität (vCPUs) nach Bedarf reduzieren.

Bewährte Methoden zur Skalierung finden Sie unterSkalierung elastischer Cluster.

> **Note**
>
> Skalierung auf Clusterebene ist ebenfalls verfügbar. Weitere Informationen finden Sie unter Skalierung von Amazon DocumentDB-Clustern.

Elastische Cluster-Zuverlässigkeit

Amazon DocumentDB ist darauf ausgelegt, zuverlässig, robust und fehlertolerant zu sein. Um die Verfügbarkeit zu verbessern, stellen Elastic Cluster zwei Knoten pro Shard bereit, die in verschiedenen Availability Zones platziert sind. Amazon DocumentDB umfasst mehrere automatische Funktionen, die es zu einer zuverlässigen Datenbanklösung machen. Weitere Informationen finden Sie unter Zuverlässigkeit von Amazon DocumentDB.

Elastischer Cluster-Speicher und Verfügbarkeit

Amazon DocumentDB DocumentDB-Daten werden in einem Cluster-Volume gespeichert, bei dem es sich um ein einzelnes virtuelles Volume handelt, das Solid-State-Laufwerke (SSDs) verwendet. Ein Cluster-Volume besteht aus sechs Kopien Ihrer Daten, die automatisch über mehrere Availability Zones in einer einzigen AWS Region repliziert werden. Diese Replikation trägt dazu bei, dass Ihre Daten sehr langlebig sind und weniger Datenverlust möglich ist. Sie trägt außerdem dazu bei, dass Ihr Cluster während eines Failovers besser verfügbar ist, da Kopien Ihrer Daten bereits in anderen Availability Zones vorhanden sind. Weitere Informationen zu Speicher, Hochverfügbarkeit und Replikation finden Sie unterAmazon DocumentDB: So funktioniert's.

Funktionale Unterschiede zwischen Amazon DocumentDB 4.0 und Elastic Clusters

Die folgenden funktionalen Unterschiede bestehen zwischen Amazon DocumentDB 4.0 und Elastic Clusters.

- Die Ergebnisse von `top` und `collStats` sind nach Shards partitioniert. Bei Datensammlungen mit mehreren Partitionen werden die Daten auf mehrere Partitionen verteilt, und die `collStats` Berichte werden aus den Partitionen aggregiert. `collScans`
- Sammlungsstatistiken von `top` und `collStats` für Sammlungen mit Sharding werden zurückgesetzt, wenn die Anzahl der Cluster-Shards geändert wird.
- Die integrierte Backup-Rolle unterstützt jetzt. `serverStatus` Aktion — Entwickler und Anwendungen mit Backup-Rolle können Statistiken über den Status des Amazon DocumentDB-Clusters sammeln.
- Das `SecondaryDelaySecs` Feld ersetzt `slaveDelay` in der `replSetGetConfig` Ausgabe.
- Der `hello` Befehl ersetzt `isMaster` — `hello` gibt ein Dokument zurück, das die Rolle des elastischen Clusters beschreibt.
- Der `$elemMatch` Operator in elastischen Clustern entspricht nur Dokumenten in der ersten Verschachtelungsebene eines Arrays. In Amazon DocumentDB 4.0 durchläuft der Operator alle Ebenen, bevor er übereinstimmende Dokumente zurücksendet. Zum Beispiel:

```
db.foo.insert(
[
    {a: {b: 5}},
    {a: {b: [5]}},
    {a: {b: [3, 7]}},
    {a: [{b: 5}]},
    {a: [{b: 3}, {b: 7}]},
    {a: [{b: [5]}]},
    {a: [{b: [3, 7]}]},
    {a: [[{b: 5}]]},
    {a: [[{b: 3}, {b: 7}]]},
    {a: [[{b: [5]}]]},
    {a: [[{b: [3, 7]}]]}
]);
// Elastic clusters
```

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

// Docdb 4.0: traverse more than one level deep
> db.foo.find({a: {$elemMatch: {b: {$elemMatch: {$lt: 6, $gt: 4}}}}}, {_id: 0})
{ "a" : [ { "b" : [ 5 ] } ] }
{ "a" : [ [ { "b" : [ 5 ] } ] ] }
```

- Die Projektion „$“ in Amazon DocumentDB 4.0 gibt alle Dokumente mit allen Feldern zurück. Bei elastischen Clustern gibt der `find` Befehl mit einer „$“ -Projektion Dokumente zurück, die dem Abfrageparameter entsprechen und nur das Feld enthalten, das der Projektion „$“ entspricht.
- In elastischen Clustern geben die `find` Befehle mit `$regex` und `$options` Abfrageparametern einen Fehler zurück: „Optionen können nicht sowohl in $regex als auch in $options festgelegt werden.“

- Bei elastischen Clustern wird `$indexOfCP` jetzt „-1" zurückgegeben, wenn:
  - Die Teilzeichenfolge wurde nicht in der `string expression` Datei gefunden, oder
  - `start`ist eine Zahl größer als`end`, oder
  - `start`ist eine Zahl, die größer als die Bytelänge der Zeichenfolge ist.

  Gibt in Amazon DocumentDB 4.0 „0" `$indexOfCP` zurück, wenn die `start` Position eine Zahl ist, die größer als `end` oder die Bytelänge der Zeichenfolge ist.
- Bei elastischen Clustern geben Projektionsoperationen z. B. Dokumente zurück`{"_id.nestedField" : 1}`, die nur das projizierte Feld enthalten. `_id fields` In Amazon DocumentDB 4.0 filtern Befehle zur verschachtelten Feldprojektion derweil kein Dokument heraus.

Erste Schritte mit Amazon DocumentDB Elastic Clusters

In diesem Abschnitt „Erste Schritte“ erfahren Sie, wie Sie Ihren ersten Elastic Cluster erstellen und abfragen können.

Es gibt viele Möglichkeiten, eine Verbindung herzustellen und mit Amazon DocumentDB zu beginnen. Das folgende Verfahren ist der schnellste, einfachste und einfachste Weg für Benutzer, mit der Nutzung unserer leistungsstarken Dokumentendatenbank zu beginnen. In diesem Handbuch wird verwendet AWS CloudShell, um eine Verbindung herzustellen und Ihren Amazon DocumentDB-

Cluster direkt vom AWS Management Console abzufragen. Neukunden, die Anspruch auf das AWS kostenlose Kontingent haben, können Amazon DocumentDB CloudShell kostenlos nutzen. Wenn Ihre AWS CloudShell Umgebung oder Ihr Amazon DocumentDB Elastic Cluster Ressourcen nutzt, die über das kostenlose Kontingent hinausgehen, werden Ihnen die normalen AWS Tarife für diese Ressourcen berechnet. Mit diesem Handbuch können Sie in weniger als 5 Minuten mit Amazon DocumentDB beginnen.

Themen

- Voraussetzungen
- Schritt 1: Erstellen Sie einen elastischen Cluster
- Schritt 2: Connect zu Ihrem Elastic Cluster her
- Schritt 3: Teilen Sie Ihre Sammlung, fügen Sie Daten ein und fragen Sie sie ab
- Schritt 4: Erkunden

Voraussetzungen

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

Erstellen Sie ein Amazon Web Services (AWS) -Konto

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

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

Um sich für eine anzumelden AWS-Konto

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

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer

Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

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

Für den Zugriff auf die Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen wie Clustern, Instances und Cluster-Parametergruppen sind Anmeldeinformationen erforderlich, mit denen Sie Ihre Anfragen authentifizieren AWS können. Weitere Informationen finden Sie unter Identity and Access Management für Amazon DocumentDB.

1. Geben Sie in der Suchleiste von IAM ein und wählen Sie im Drop-down-Menü IAM aus. AWS Management Console
2. Sobald Sie sich in der IAM-Konsole befinden, wählen Sie im Navigationsbereich Benutzer aus.
3. Wählen Sie Ihren Nutzernamen aus.
4. Klicken Sie auf Berichtigung hinzufügen.
5. Wählen Sie Richtlinien direkt anhängen aus.
6. Geben Sie `AmazonDocDBElasticFullAccess` in die Suchleiste ein und wählen Sie sie aus, sobald sie in den Suchergebnissen erscheint.
7. Klicken Sie auf Weiter.
8. Klicken Sie auf Berichtigung hinzufügen.

> **Note**
>
> Ihr AWS Konto enthält in jeder Region eine Standard-VPC. Wenn Sie sich für die Verwendung einer Amazon VPC entscheiden, führen Sie die Schritte im Thema Amazon VPC erstellen im Amazon VPC-Benutzerhandbuch aus.

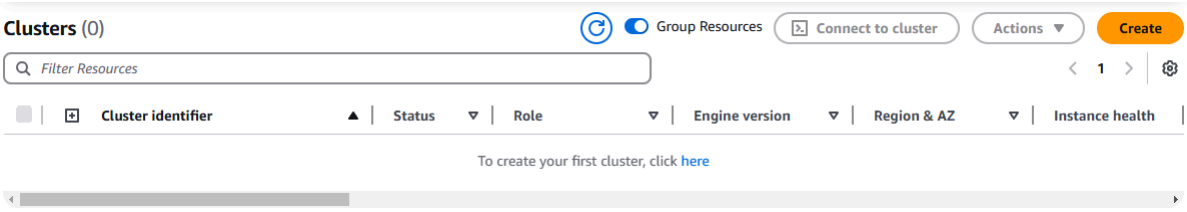
Schritt 1: Erstellen Sie einen elastischen Cluster

In diesem Abschnitt erklären wir, wie Sie einen brandneuen elastischen Cluster erstellen, indem Sie entweder das AWS Management Console oder AWS CLI verwenden und die folgenden Anweisungen verwenden.

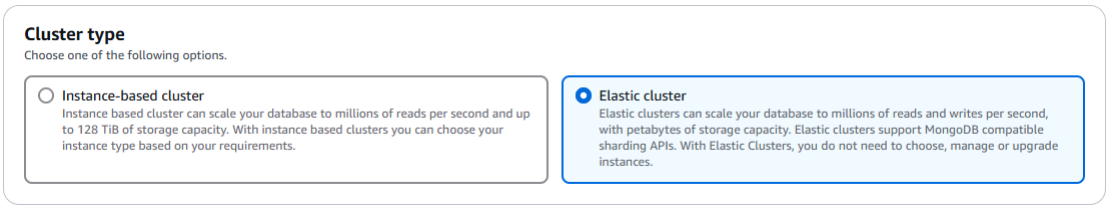
Using the AWS Management Console

Um eine elastische Cluster-Konfiguration zu erstellen, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie in der Amazon DocumentDB Management Console unter Clusters die Option Create aus.



3. Wählen Sie auf der Seite Amazon DocumentDB-Cluster erstellen im Abschnitt Clustertyp die Option Elastic Cluster aus.



4. Konfigurieren Sie im Abschnitt Konfiguration Folgendes:

   a. Geben Sie im Feld Clustername eine eindeutige Cluster-ID ein (gemäß den Benennungsanforderungen unter dem Feld).

   b. Geben Sie im Feld Anzahl der Shards die Anzahl der Shards ein, die Sie in Ihrem Cluster haben möchten. Die maximale Anzahl von Shards pro Cluster ist 32.

   > **Note**
   >
   > Für jeden Shard werden zwei Knoten bereitgestellt. Beide Knoten werden dieselbe Shard-Kapazität haben.

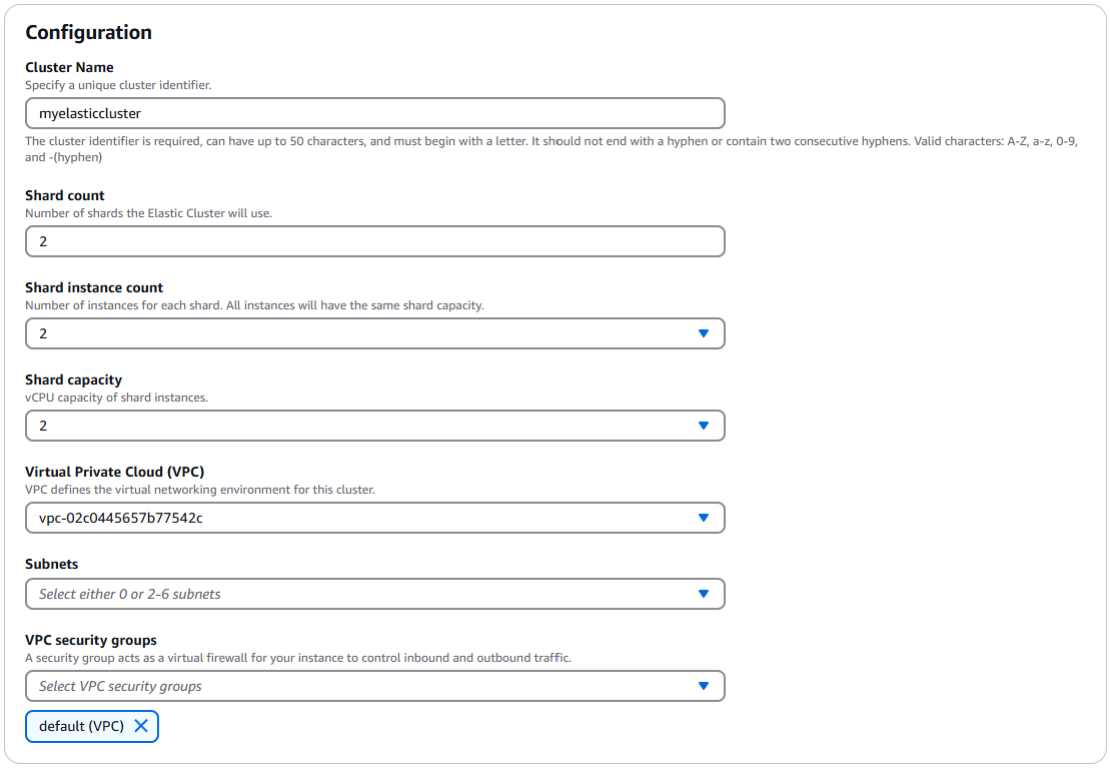
   c. Wählen Sie im Feld Anzahl der Shard-Instanzen die Anzahl der Replikatinstanzen aus, die Sie jedem Shard zuordnen möchten. Die maximale Anzahl von Shard-Instanzen beträgt 16, in Schritten von 1. Alle Replikatinstanzen haben dieselbe Shard-Kapazität, wie im folgenden Feld definiert. Zu Testzwecken sollte der Standardwert 2 ausreichen.

> **Note**
>
> Die Anzahl der Replikatinstanzen gilt für alle Shards im Elastic Cluster. Ein Wert für die Anzahl der Shard-Instances von 1 bedeutet, dass es eine Writer-Instance gibt, und alle weiteren Instances sind Replikate, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können. Zu Testzwecken sollte der Standardwert 2 ausreichen.

d. Wählen Sie im Feld Shard-Kapazität die Anzahl der virtuellen CPUs (vCPUs) aus, die Sie jeder Shard-Instance zuordnen möchten. Die maximale Anzahl von v CPUs pro Shard-Instanz ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64. Zu Testzwecken sollte der Standardwert 2 ausreichen.

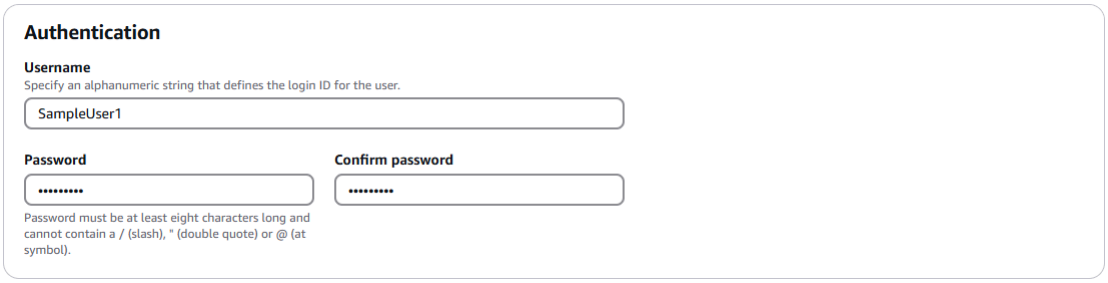
e. Wählen Sie im Feld Virtual Private Cloud (VPC) eine VPC aus der Dropdownliste aus.

f. Für Subnetze und VPC-Sicherheitsgruppen können Sie die Standardeinstellungen verwenden oder drei Subnetze Ihrer Wahl und bis zu drei VPC-Sicherheitsgruppen (mindestens eine) auswählen.



5. Geben Sie im Abschnitt Authentifizierung im Feld Benutzername eine Zeichenfolge ein, die den Anmeldenamen des Hauptbenutzers identifiziert.

Geben Sie im Feld Passwort ein eindeutiges Passwort ein, das den Anweisungen entspricht, und bestätigen Sie es anschließend.



6. Behalten Sie im Abschnitt Verschlüsselung die Standardeinstellung (Standardschlüssel) bei.

   Optional können Sie einen von Ihnen erstellten AWS KMS key ARN eingeben. Weitere Informationen finden Sie unter Datenverschlüsselung im Ruhezustand für elastische Amazon DocumentDB-Cluster.

   > **Important**
   >
   > Die Verschlüsselung muss für elastische Cluster aktiviert sein.

7. Bearbeiten Sie im Abschnitt Backup die Felder entsprechend Ihren Backup-Anforderungen. Zu Testzwecken können Sie die Standardeinstellungen beibehalten.



   a. Aufbewahrungszeitraum für Backup — Wählen Sie in der Liste die Anzahl der Tage aus, für die automatische Backups dieses Clusters aufbewahrt werden sollen, bevor sie gelöscht werden.

   b. Backup-Fenster — Legen Sie die tägliche Uhrzeit und Dauer fest, während der Amazon DocumentDB Backups dieses Clusters erstellen soll.

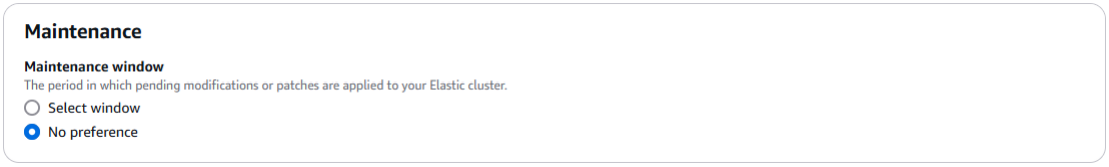
      i. Wählen Sie Fenster auswählen, wenn Sie die Uhrzeit und Dauer der Erstellung von Backups konfigurieren möchten.

Startzeit — Wählen Sie in der ersten Liste die Startzeit (UTC) für den Start Ihrer automatischen Backups aus. Wählen Sie in der zweiten Liste die Minute für den Beginn der automatischen Backups aus.

Dauer — Wählen Sie in der Liste die Anzahl der Stunden aus, die für die Erstellung automatischer Backups reserviert werden sollen.

ii. Wählen Sie Keine Präferenz, wenn Amazon DocumentDB den Zeitpunkt und die Dauer der Erstellung von Backups wählen soll.

8. Wählen Sie im Abschnitt Wartung den Tag, die Uhrzeit und die Dauer aus, an dem Änderungen oder Patches auf Ihren Cluster angewendet werden. Zu Testzwecken können Sie die Standardeinstellungen beibehalten.



9. Wählen Sie Cluster erstellen.

Der elastische Cluster wird jetzt bereitgestellt. Es kann bis zu einigen Minuten dauern, bis dieser Vorgang abgeschlossen ist. Sie können eine Verbindung zu Ihrem Cluster herstellen, wenn der Elastic Cluster-Status in der Cluster-Liste als Verfügbar angezeigt wird.

Using the AWS CLI

Um einen elastischen Cluster mit dem zu erstellen AWS CLI, verwenden Sie den `create-cluster` Vorgang mit den folgenden Parametern:

- --cluster-name—Erforderlich. Der aktuelle Name des Elastic Scale Clusters, wie er bei der Erstellung eingegeben oder zuletzt geändert wurde.
- --shard-capacity—Erforderlich. Die Anzahl der V, die jedem Shard CPUs zugewiesen wurden. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.
- --shard-count—Erforderlich. Die Anzahl der dem Cluster zugewiesenen Shards. Das Maximum ist 32.
- --shard-instance-count— Fakultativ. Die Anzahl der Replikatinstanzen, die für alle Shards in diesem Cluster gelten. Das Maximum ist 16.
- --admin-user-name—Erforderlich. Der dem Admin-Benutzer zugeordnete Benutzername.
- --admin-user-password—Erforderlich. Das dem Admin-Benutzer zugeordnete Passwort.

- --auth-type—Erforderlich. Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das für den Zugriff auf den Elastic Cluster verwendete Passwort abgerufen werden soll. Gültige Typen sind `PLAIN_TEXT` oder`SECRET_ARN`.
- --vpc-security-group-ids— Fakultativ. Konfigurieren Sie eine Liste von EC2 VPC-Sicherheitsgruppen, die diesem Cluster zugeordnet werden sollen.
- --preferred-maintenance-window— Fakultativ. Konfigurieren Sie den wöchentlichen Zeitraum, in dem die Systemwartung erfolgen kann, in UTC (Universal Coordinated Time).

  Das Format ist:`ddd:hh24:mi-ddd:hh24:mi`. Gültige Tage (ddd): Mo, Di, Mi, Do, Fr, Sa, So

  Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock für jede Amazon Web Services Services-Region ausgewählt wird und an einem zufälligen Wochentag stattfindet.

  Zeitfenster von mindestens 30 Minuten.
- --kms-key-id— Fakultativ. Konfigurieren Sie die KMS-Schlüssel-ID für einen verschlüsselten Cluster.

  Die KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon Web Services Services-Konto erstellen, dem der KMS-Verschlüsselungsschlüssel gehört, der zur Verschlüsselung des neuen Clusters verwendet wird, können Sie den KMS-Schlüsselalias anstelle des ARN für den KMS-Verschlüsselungsschlüssel verwenden.

  Wenn in kein Verschlüsselungsschlüssel angegeben ist KmsKeyId und der `StorageEncrypted` Parameter wahr ist, verwendet Amazon DocumentDB Ihren Standard-Verschlüsselungsschlüssel.
- --preferred-backup-window— Optional. Der bevorzugte tägliche Zeitraum, in dem automatische Backups erstellt werden. Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils AWS-Region 8 Stunden ausgewählt wird.
- --backup-retention-period— Fakultativ. Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Der Standardwert lautet 1.
- --storage-encrypted—Fakultativ. Konfiguriert, ob der Cluster verschlüsselt oder nicht verschlüsselt ist.

  `--no-storage-encrypted`gibt an, dass der Cluster nicht verschlüsselt ist.
- --subnet-ids— Fakultativ. Konfigurieren Sie Netzwerk-Subnetz-IDs.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch Ihre eigenen Informationen.

> **Note**
>
> Die folgenden Beispiele beinhalten die Erstellung eines bestimmten KMS-Schlüssels. Um den Standard-KMS-Schlüssel zu verwenden, schließen Sie den `--kms-key-id` Parameter nicht ein.

Für Linux, macOS oder Unix:

```
aws docdb-elastic create-cluster \
     --cluster-name sample-cluster-123 \
     --shard-capacity 8 \
     --shard-count 4 \
     --shard-instance-count 3 \
     --auth-type PLAIN_TEXT \
     --admin-user-name testadmin \
     --admin-user-password testPassword \
     --vpc-security-group-ids ec-65f40350 \
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \
     --preferred-backup-window 18:00-18:30 \
     --backup-retention-period 7
```

Für Windows:

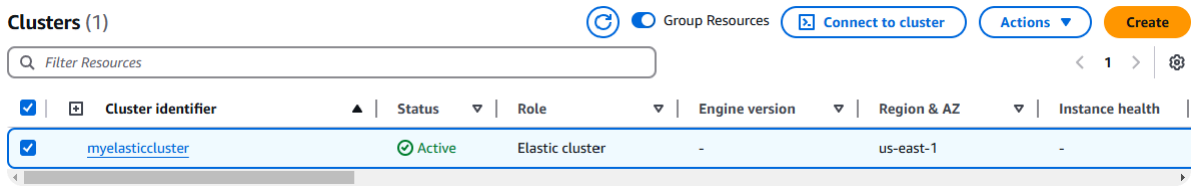
```
aws docdb-elastic create-cluster ^
     --cluster-name sample-cluster-123 ^
     --shard-capacity 8 ^
     --shard-count 4 ^
     --shard-instance-count 3 ^
     --auth-type PLAIN_TEXT ^
     --admin-user-name testadmin ^
     --admin-user-password testPassword ^
     --vpc-security-group-ids ec-65f40350 ^
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9 \
```

```
    --preferred-backup-window 18:00-18:30 \
    --backup-retention-period 7
```

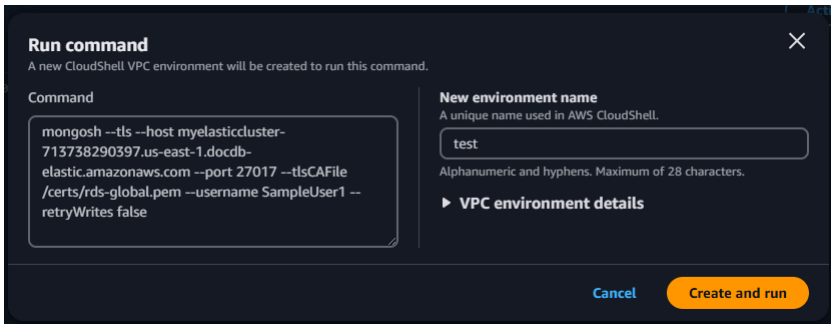
Schritt 2: Connect zu Ihrem Elastic Cluster her

Stellen Sie mithilfe AWS CloudShell von... Connect zu Ihrem Amazon DocumentDB Elastic Cluster her.

1. Suchen Sie auf der Amazon DocumentDB-Managementkonsole unter Clusters den Elastic Cluster, den Sie erstellt haben. Wählen Sie Ihren Cluster aus, indem Sie auf das Kontrollkästchen neben dem Cluster klicken.



2. Klicken Sie neben dem Dropdownmenü Aktionen auf Mit Cluster Connect. Diese Schaltfläche ist erst aktiviert, wenn Sie auf das Kontrollkästchen neben Ihrem Cluster geklickt haben und der Status des Clusters als Verfügbar angezeigt wird. Der Befehlsbildschirm CloudShell Ausführen wird angezeigt.
3. Geben Sie im Feld Neuer Umgebungsname einen eindeutigen Namen ein, z. B. „Test“, und klicken Sie auf Erstellen und ausführen. VPC-Umgebungsdetails werden automatisch für Ihre Amazon DocumentDB DocumentDB-Datenbank konfiguriert.



4. Wenn Sie dazu aufgefordert werden, geben Sie das Passwort ein, das Sie in Schritt 1: Amazon DocumentDB Elastic Cluster erstellen (Unterschritt 5) erstellt haben.



Nachdem Sie Ihr Passwort eingegeben haben und Ihre Aufforderung lautet: Sie sind erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden `direct: mongos] <env-name>>`

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

Schritt 3: Teilen Sie Ihre Sammlung, fügen Sie Daten ein und fragen Sie sie ab

Elastische Cluster bieten Unterstützung für Sharding in Amazon DocumentDB. Jetzt, da Sie mit Ihrem Cluster verbunden sind, können Sie den Cluster teilen, Daten einfügen und einige Abfragen ausführen.

1. Um eine Sammlung zu teilen, geben Sie Folgendes ein:

```
sh.shardCollection("db.Employee1" , { "Employeeid" : "hashed" })
```

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

```
db.Employee1.insertOne({"Employeeid":1, "Name":"Joe", "LastName": "Bruin", "level":
 1 })
```

   Die folgende Ausgabe wird angezeigt:

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

3. Um das Dokument zu lesen, das Sie geschrieben haben, geben Sie den `findOne()` Befehl ein (es wird ein einzelnes Dokument zurückgegeben):

```
db.Employee1.findOne()
```

Die folgende Ausgabe wird angezeigt:

```
{
"_id" : ObjectId("61f344e0594fe1a1685a8151"),
"EmployeeID" : 1,
"Name" : "Joe",
"LastName" : "Bruin",
"level" : 1
}
```

4. Um ein paar weitere Abfragen durchzuführen, sollten Sie einen Anwendungsfall für ein Spieleprofil in Betracht ziehen. Fügen Sie zunächst einige Einträge in eine Sammlung mit dem Titel „Mitarbeiter“ ein. Geben Sie Folgendes ein:

```
db.profiles.insertMany([ { "_id": 1, "name": "Matt", "status": "active", "level":
 12, "score": 202 },
     { "_id": 2, "name": "Frank", "status": "inactive", "level": 2, "score": 9 },
     { "_id": 3, "name": "Karen", "status": "active", "level": 7, "score": 87 },
     { "_id": 4, "name": "Katie", "status": "active", "level": 3, "score": 27 }
])
```

Die folgende Ausgabe wird angezeigt:

```
{ acknowledged: true,
     insertedIds: {
        '0': ObjectId('679d02cd6b5a0581be78bcbd'),
        '1': ObjectId('679d02cd6b5a0581be78bcbe'),
        '2': ObjectId('679d02cd6b5a0581be78bcbf'),
        '3': ObjectId('679d02cd6b5a0581be78bcc0')
    }
}
```

5. Um alle Dokumente in der Profilsammlung zurückzugeben, geben Sie den Befehl `find` () ein:

```
db.Employee.find()
```

Die Daten, die Sie in Schritt 4 eingegeben haben, werden angezeigt.

6. Um ein einzelnes Dokument abzufragen, fügen Sie einen Filter hinzu (z. B.: „Katie“). Geben Sie Folgendes ein:

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

Die folgende Ausgabe wird angezeigt:

```
[

   {

     _id: ObjectId('679d02cd6b5a0581be78bcc0'),

     Employeeid: 4,

     name: 'Katie',

     lastname: 'Schaper',

     level: 3

   }

]
```

7. Um ein Profil zu finden und es zu ändern, geben Sie den `findAndModify` Befehl ein. In diesem Beispiel erhält der Mitarbeiter „Matt“ eine höhere Stufe von „14":

```
db.Employee.findAndModify({
    query: { "Employeeid" : 1, "name" : "Matt"},
    update: { "Employeeid" : 1, "name" : "Matt", "lastname" : "Winkle", "level" :
 14 }
})
```

Die folgende Ausgabe wird angezeigt (beachten Sie, dass sich der Pegel noch nicht geändert hat):

```
{
```

```
   _id: ObjectId('679d02cd6b5a0581be78bcbd'),

   Employeeid: 1,

   name: 'Matt',

   lastname: 'Winkle',

   level: 12

}
```

8. Geben Sie die folgende Abfrage ein, um den Levelanstieg zu überprüfen:

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

Die folgende Ausgabe wird angezeigt:

```
[
   {

   _id: ObjectId('679d02cd6b5a0581be78bcbd'),

   Employeeid: 1,

   name: 'Matt',

   lastname: 'Winkle',

   level: 14

   }
]
```

Schritt 4: Erkunden

Herzlichen Glückwunsch! Sie haben das Verfahren Erste Schritte für Amazon DocumentDB Elastic Clusters erfolgreich abgeschlossen.

Was kommt als Nächstes? Erfahren Sie, wie Sie diese Datenbank mit einigen ihrer beliebten Funktionen voll ausschöpfen können:

- Bewährte Methoden für Amazon DocumentDB Elastic Cluster
- Verwaltung elastischer Amazon DocumentDB-Cluster

> **Note**
>
> Für den Elastic Cluster, den Sie im Rahmen dieser ersten Schritte erstellt haben, fallen weiterhin Kosten an, sofern Sie ihn nicht löschen. Eine Anleitung finden Sie unter Löschen eines elastischen Clusters.

Bewährte Methoden für Amazon DocumentDB Elastic Cluster

Lernen Sie bewährte Methoden für die Arbeit mit elastischen Amazon DocumentDB-Clustern kennen. Alle bewährten Methoden für instanzbasierte Amazon DocumentDB-Cluster gelten auch für elastische Cluster. Dieser Abschnitt wird fortlaufend aktualisiert, wenn neue bewährte Methoden identifiziert werden.

Themen

- Auswahl von Shard-Schlüsseln
- Verbindungsverwaltung
- Sammlungen, die nicht geteilt wurden
- Skalierung elastischer Cluster
- Überwachung elastischer Cluster

Auswahl von Shard-Schlüsseln

In der folgenden Liste werden Richtlinien für die Erstellung von Shard-Schlüsseln beschrieben.

- Verwenden Sie einen gleichmäßig verteilten Hashschlüssel, um Ihre Daten auf alle Shards in Ihrem Cluster zu verteilen (vermeiden Sie Hotkeys).
- Verwenden Sie Ihren Shard-Schlüssel in allen read/update/delete Anfragen, um Scatter-Gather-Abfragen zu vermeiden.
- Vermeiden Sie verschachtelte Shard-Schlüssel, wenn Sie Operationen ausführen. read/update/delete

- Wenn Sie Batch-Operationen ausführen, sollten Sie `ordered` den Wert auf False setzen, damit alle Shards parallel ausgeführt werden können und die Latenzen verbessert werden.

Verbindungsverwaltung

In der folgenden Liste werden Richtlinien für die Verwaltung Ihrer Verbindungen zu Ihrer Datenbank beschrieben.

- Überwachen Sie die Anzahl Ihrer Verbindungen und wie häufig neue Verbindungen geöffnet und geschlossen werden.
- Verteilen Sie Ihre Verbindungen auf alle Subnetze in der Konfiguration Ihrer Anwendung. Wenn Ihr Cluster in mehreren Subnetzen konfiguriert ist, Sie aber nur eine Teilmenge der Subnetze nutzen, kann es zu Engpässen bei der maximalen Anzahl von Verbindungen kommen.

Sammlungen, die nicht geteilt wurden

Im Folgenden wird eine Richtlinie für Sammlungen ohne Sharding beschrieben.

- Versuchen Sie bei der Arbeit mit Sammlungen ohne Sharded, um die Last zu verteilen, häufig genutzte Sammlungen ohne Sharded in verschiedenen Datenbanken zu speichern. Elastische Amazon DocumentDB-Cluster platzieren Datenbanken auf verschiedenen Shards und legen Sammlungen ohne Shards für dieselbe Datenbank auf demselben Shard zusammen.

Skalierung elastischer Cluster

In der folgenden Liste werden Richtlinien für die Skalierung Ihrer elastischen Cluster beschrieben.

- Skalierungsvorgänge können für einen kurzen Zeitraum zu zeitweiligen Datenbank- und Netzwerkfehlern führen. Vermeiden Sie nach Möglichkeit eine Skalierung zu Spitzenzeiten. Versuchen Sie, während der Wartungsfenster zu skalieren.
- Das Hoch- und Herunterskalieren der Shard-Kapazität (Änderung der vCPU-Anzahl pro Shard) zur Erhöhung der Rechenleistung wird dem Erhöhen oder Verringern der Shard-Anzahl vorgezogen, da dies schneller ist und eine kürzere Dauer von zeitweiligen Datenbank- und Netzwerkfehlern hat.
- Wenn Sie ein Wachstum erwarten, sollten Sie die Anzahl der Shards erhöhen, anstatt die Shard-Kapazität zu skalieren. Auf diese Weise können Sie Ihren Cluster skalieren, indem Sie die Shard-Kapazität für Szenarien erhöhen, in denen Sie schnell skalieren müssen.

- Überwachen Sie Ihre clientseitigen Wiederholungsrichtlinien und versuchen Sie es erneut mit exponentiellem Backoff und Jitter, um eine Überlastung Ihrer Datenbank zu vermeiden, wenn bei der Skalierung Fehler auftreten.

Überwachung elastischer Cluster

In der folgenden Liste werden Richtlinien für die Überwachung Ihrer elastischen Cluster beschrieben.

- Verfolgen Sie das peak-to-average Verhältnis Ihrer Metriken pro Shard, um festzustellen, ob Sie ungleichmäßigen Traffic generieren (Sie haben einen Hotkey/Hotspot). Die wichtigsten Kennzahlen zur Erfassung von Kennzahlen sind: peak-to-average
  - `PrimaryInstanceCPUUtilization`
    - Dies kann auf der Ebene pro Shard überwacht werden.
    - Auf Clusterebene können Sie den Mittelwert bis zur Abweichung von p99 überwachen.
  - `PrimaryInstanceFreeableMemory`
    - Dies kann auf der Ebene pro Shard überwacht werden.
    - Auf Clusterebene können Sie den Mittelwert bis zur Abweichung von p99 überwachen.
  - `DatabaseCursorsMax`
    - Dies sollte auf der Ebene pro Shard überwacht werden, um die Verzerrung zu ermitteln.
  - `Documents-Inserted/Updated/Returned/Deleted`
    - Dies sollte auf der Ebene pro Shard überwacht werden, um die Verzerrung zu ermitteln.

Verwaltung elastischer Amazon DocumentDB-Cluster

Um einen elastischen Amazon DocumentDB-Cluster zu verwalten, benötigen Sie eine IAM-Richtlinie mit den entsprechenden Berechtigungen für die Amazon DocumentDB-Steuerebene. Diese Berechtigungen ermöglichen es Ihnen, Cluster zu erstellen, zu ändern und zu löschen. Die Amazon Document DBFull Access-Richtlinie bietet alle erforderlichen Berechtigungen für die Verwaltung eines elastischen Amazon DocumentDB-Clusters.

Die folgenden Themen zeigen, wie Sie verschiedene Aufgaben bei der Arbeit mit elastischen Amazon DocumentDB-Clustern ausführen.

Themen

- Modifizieren elastischer Cluster-Konfigurationen

- Überwachen eines elastischen Clusters
- Löschen eines elastischen Clusters
- Verwaltung von Elastic Cluster-Snapshots
- Stoppen und Starten eines elastischen Amazon DocumentDB-Clusters
- Wartung elastischer Amazon DocumentDB-Cluster

Modifizieren elastischer Cluster-Konfigurationen

In diesem Abschnitt wird anhand der folgenden Anweisungen erklärt, wie Elastic Cluster entweder AWS CLI mit AWS Management Console oder geändert werden.

Eine Änderung des Clusters wird hauptsächlich zur Skalierung von Shards verwendet, indem die Anzahl der Shards und/oder die Shard-Rechenkapazität erhöht oder verringert werden.

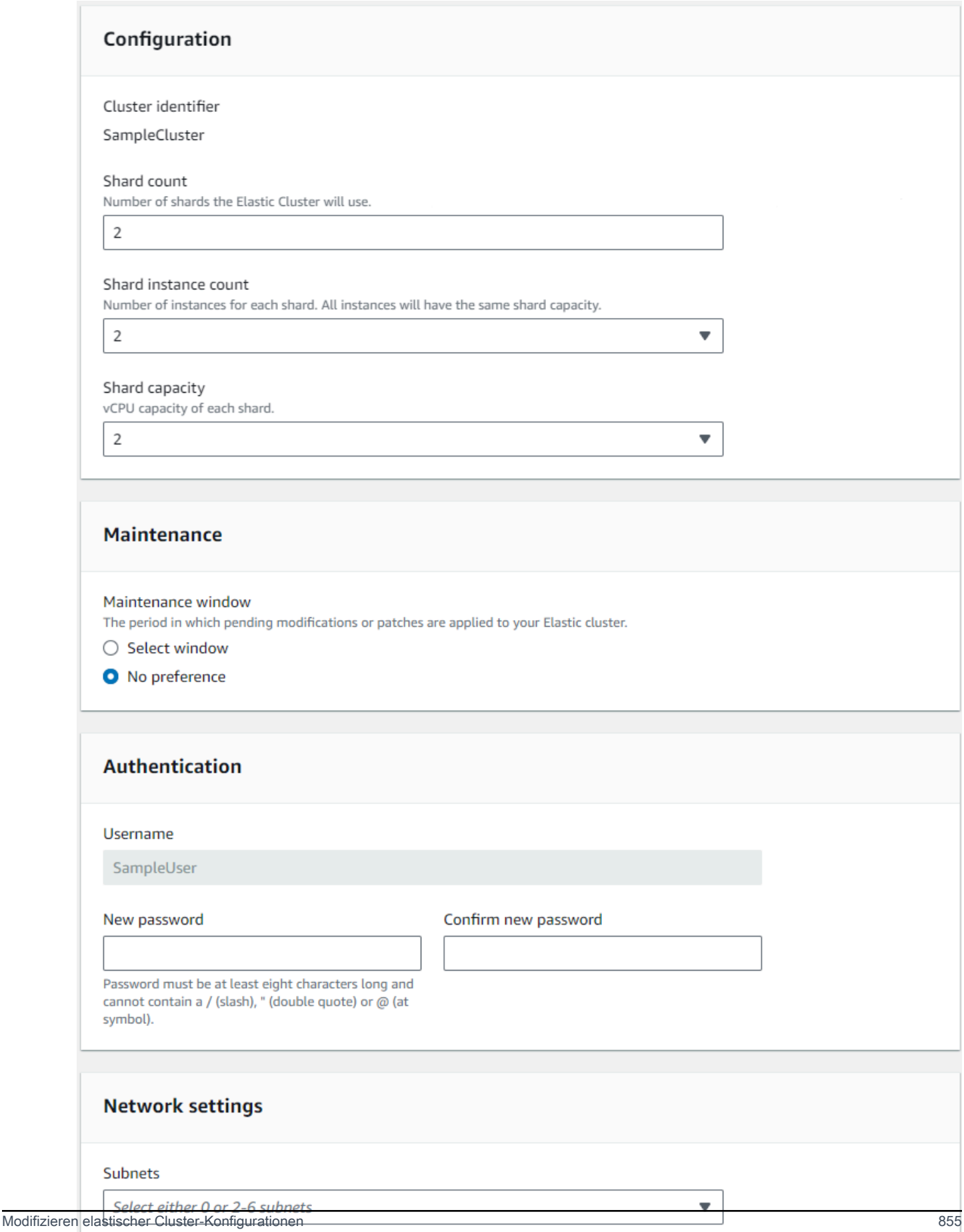
Using the AWS Management Console

Um eine elastische Cluster-Konfiguration zu ändern, verwenden Sie: AWS Management Console

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Klicken Sie im Navigationsbereich auf Cluster.

   > **Tip**
   >
   > Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Cluster-ID den Namen des Clusters aus, den Sie ändern möchten.
4. Wählen Sie Ändern aus.
5. Bearbeiten Sie die Felder, die Sie ändern möchten, und wählen Sie dann Cluster ändern aus.


Configuration
Cluster identifier
SampleCluster
Shard count
Number of shards the Elastic Cluster will use.
2
Shard instance count
Number of instances for each shard. All instances will have the same shard capacity.
2
Shard capacity
vCPU capacity of each shard.
2
Maintenance
Maintenance window
The period in which pending modifications or patches are applied to your Elastic cluster.
Select window
No preference
Authentication
Username
SampleUser
New password
Confirm new password
Password must be at least eight characters long and cannot contain a / (slash), " (double quote) or @ (at symbol).
Network settings
Subnets
Select either 0 or 2-6 subnets
subnet-0b2962f92a0f5a8fb
subnet-08c6d849efd4dfe96
VPC security groups

> **Note**
>
> Alternativ können Sie auf das Dialogfeld „Cluster ändern“ zugreifen, indem Sie auf der Seite Cluster das Kästchen neben Ihrem Cluster markieren und Aktionen und dann Ändern auswählen.

Using the AWS CLI

Um eine elastische Cluster-Konfiguration mithilfe von zu ändern AWS CLI, verwenden Sie den `update-cluster` Vorgang mit den folgenden Parametern:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, den Sie ändern möchten.
- **`--shard-capacity`**— Fakultativ. Die Anzahl von v, die jedem Shard CPUs zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.
- **`--shard-count`**— Fakultativ. Die Anzahl der dem Cluster zugewiesenen Shards. Das Maximum ist 32.
- **`--shard-instance`**-Anzahl — optional. Die Anzahl der Replikatinstanzen, die für alle Shards in diesem Cluster gelten. Das Maximum ist 16.
- **`--auth-type`**— Fakultativ. Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das für den Zugriff auf den Elastic Cluster verwendete Passwort abgerufen werden soll. Gültige Typen sind `PLAIN_TEXT` oder`SECRET_ARN`.
- **`--admin-user-password`**— Fakultativ. Das dem Admin-Benutzer zugeordnete Passwort.
- **`--vpc-security-group-ids`**— Fakultativ. Konfigurieren Sie eine Liste von Amazon EC2 - und Amazon Virtual Private Cloud (VPC) -Sicherheitsgruppen, die diesem Cluster zugeordnet werden sollen.
- **`--preferred-maintenance-window`**— Optional. Konfigurieren Sie den wöchentlichen Zeitraum, in dem die Systemwartung erfolgen kann, in UTC (Universal Coordinated Time)

  Das Format ist:`ddd:hh24:mi-ddd:hh24:mi`. Gültige Tage (ddd): Mo, Di, Mi, Do, Fr, Sa, So

  Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock für jede Amazon Web Services Services-Region ausgewählt wird und an einem zufälligen Wochentag stattfindet.

  Zeitfenster von mindestens 30 Minuten.
- **`--subnet-ids`**— Fakultativ. Konfigurieren Sie Netzwerk-Subnetz-IDs.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic update-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
    --shard-capacity 8 \
    --shard-count 4 \
    --shard-instance-count 3 \
    --admin-user-password testPassword \
    --vpc-security-group-ids ec-65f40350 \
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Für Windows:

```
aws docdb-elastic update-cluster ^
    --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
    --shard-capacity 8 ^
    --shard-count 4 ^
    --shard-instance-count 3 ^
    --admin-user-password testPassword ^
    --vpc-security-group-ids ec-65f40350 ^
    --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Informationen zur Überwachung des Status des Elastic Clusters nach Ihrer Änderung finden Sie unter Elastic Cluster überwachen.

Überwachen eines elastischen Clusters

In diesem Abschnitt wird anhand der folgenden Anweisungen erklärt, wie Sie Ihren Elastic Cluster entweder AWS CLI mithilfe der AWS Management Console oder überwachen können.

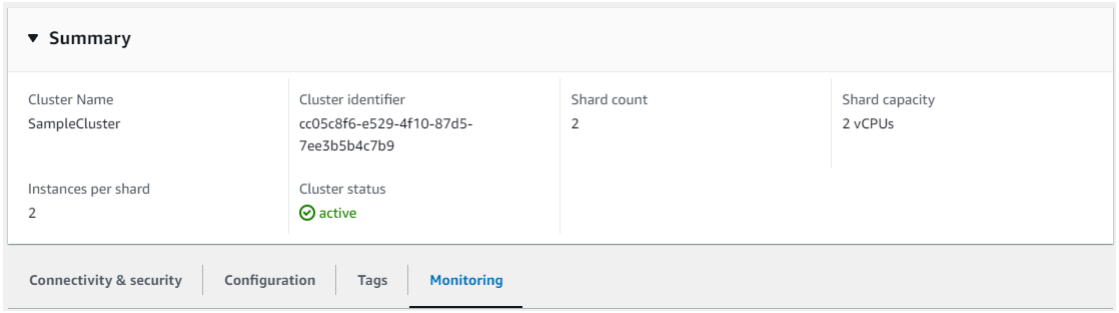
Using the AWS Management Console

Um eine elastische Cluster-Konfiguration zu überwachen, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Cluster-ID den Namen des Clusters aus, den Sie überwachen möchten.
4. Wählen Sie die Registerkarte Überwachung.



Für die folgenden Überwachungskategorien CloudWatch werden eine Reihe von Diagrammen von Amazon angezeigt:

- Nutzung der Ressourcen
- Durchsatz
- Operationen
- System (System)

Sie können auch CloudWatch über Amazon zugreifen AWS Management Console , um Ihre eigene Überwachungsumgebung für Ihre elastischen Cluster einzurichten.

Using the AWS CLI

Um eine bestimmte Elastic Cluster-Konfiguration mithilfe von zu überwachen AWS CLI, verwenden Sie den `get-cluster` Vorgang mit den folgenden Parametern:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, für den Sie Informationen benötigen.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic get-cluster \
    --cluster-arn arn:aws:docdb-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Für Windows:

```
aws docdb-elastic get-cluster ^
    --cluster-arn arn:aws:docdb:-elastic:us-west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus:

```
"cluster": {
        ...
        "clusterArn": "arn:aws:docdb-elastic:us-
west-2:123456789012:cluster:/68ffcdf8-e3af-40a3-91e4-24736f2dacc9",
        "clusterEndpoint": "stretch-11-477568257630.us-east-1.docdb-
elastic.amazonaws.com",
        "readerEndpoint": "stretch-11-477568257630-ro.us-east-1.docdb-
elastic.amazonaws.com",
        "clusterName": "stretch-11",
        "shardCapacity": 2,
        "shardCount": 3,
        "shardInstanceCount: 5,
        "status": "ACTIVE",
        ...
 }
```

Weitere Informationen finden Sie `DescribeClusterSnapshot` in der Amazon DocumentDB Resource Management API-Referenz.

Verwenden Sie den `list-clusters` Vorgang mit den folgenden Parametern AWS CLI, um die Details aller Elastic Cluster anzuzeigen, die den verwenden:

- **`--next-token`**— Optional. Wenn die Anzahl der ausgegebenen Elemente (`--max-results`) geringer als die Gesamtanzahl der Elemente ist, die von den zugrunde liegenden API-Aufrufen

zurückgeliefert werden, enthält die Ausgabe ein `NextToken`. Dieses können Sie in einem anschließenden Befehl zum Abrufen der nächsten Gruppe von Elementen übergeben.

- **`--max-results`**—Fakultativ. Die Gesamtzahl der Elemente, die in der Ausgabe des Befehls zurückgegeben werden sollen. Wenn mehr Ergebnisse als der angegebene `max-results` Wert vorhanden sind, ist ein Paginierungstoken (`next-token`) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.
  - Standard: 100
  - Mindestens 20, maximal 100

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic list-clusters \
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== \
    --max-results 2
```

Für Windows:

```
aws docdb-elastic list-clusters ^
    --next-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ== ^
    --max-results 2
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus:

```
{
   "Clusters": [
      {
         "ClusterIdentifier":"mycluster-1",
         "ClusterArn":"arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster"
         "Status":"available",
         "ClusterEndpoint":"sample-cluster.sharded-cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com"
       }
       {
         "ClusterIdentifier":"mycluster-2",
         "ClusterArn":"arn:aws:docdb:us-west-2:987654321098:sharded-cluster:sample-
cluster"
```

```
         "Status":"available",
         "ClusterEndpoint":"sample-cluster2.sharded-cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com"
       }
   ]
}
```

Löschen eines elastischen Clusters

In diesem Abschnitt erklären wir, wie Sie einen elastischen Cluster löschen, indem Sie entweder die AWS Management Console oder AWS CLI verwenden, und die folgenden Anweisungen verwenden.

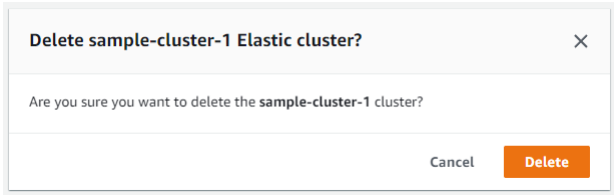
Using the AWS Management Console

Um eine elastische Cluster-Konfiguration zu löschen, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Aktivieren Sie in der Cluster-Listentabelle das Kontrollkästchen links neben dem Clusternamen, den Sie löschen möchten, und wählen Sie dann Aktionen aus. Wählen Sie im Dropdown-Menü Löschen aus.
4. Im elastischen Cluster „Clustername“ löschen? Wählen Sie im Dialogfeld Löschen aus.



Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Clusterstatus finden Sie unter Überwachen des Status eines Amazon DocumentDB-Clusters.

Using the AWS CLI

Um einen Elastic Cluster mit dem zu löschen AWS CLI, verwenden Sie den `delete-cluster` Vorgang mit den folgenden Parametern:

- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, den Sie löschen möchten.
- **`--no-skip-final-backup`**— Fakultativ. Wenn Sie eine endgültige Sicherung wünschen, müssen Sie diesen Parameter zusammen mit einem Namen für die endgültige Sicherung angeben. Sie müssen entweder `--final-backup-identifier` oder `--skip-final-backup` angeben.
- **`--skip-final-backup`**— Fakultativ. Verwenden Sie diesen Parameter nur, wenn Sie vor dem Löschen Ihres Clusters kein letztes Backup erstellen möchten. Standardmäßig wird ein letzter Snapshot erstellt.

Die folgenden AWS CLI Codebeispiele löschen einen Cluster mit dem ARN arn:aws:docdb:us-west- 2:123456789012:sharded-cluster:sample-cluster mit einer endgültigen Sicherung.

*`user input placeholder`*Ersetzen Sie im folgenden Beispiel jede durch Ihre eigenen Informationen..

Für Linux, macOS oder Unix:

```
aws docdb-elastic delete-cluster \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster \
    --no-skip-final-backup \
    --final-backup-identifier finalArnBU-arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

Für Windows:

```
aws docdb-elastic delete-cluster ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster ^
    --no-skip-final-backup ^
    --final-backup-identifier finalArnBU-arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster
```

Die folgenden AWS CLI Codebeispiele löschen einen Cluster mit dem ARN arn:aws:docdb:us-west- 2:123456789012:sharded-cluster:sample-cluster, ohne eine endgültige Sicherung zu erstellen.

*user input placeholder*Ersetzen Sie im folgenden Beispiel jedes durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic delete-cluster \
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster \
    --skip-final-backup \
```

Für Windows:

```
aws docdb-elastic delete-cluster ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-cluster ^
    --skip-final-backup ^
```

Die Ausgabe des `delete-cluster` Vorgangs ist eine Anzeige des Clusters, den Sie löschen.

Es dauert einige Minuten, bis der Cluster gelöscht ist. Informationen zur Überwachung des Clusterstatus finden Sie unter Überwachen des Status eines Amazon DocumentDB-Clusters.

Verwaltung von Elastic Cluster-Snapshots

Manuelle Snapshots können erstellt werden, nachdem ein elastischer Cluster erstellt wurde. Automatisierte Backups werden in dem Moment erstellt, in dem der Elastic Cluster-Snapshot erstellt wird.

> **Note**
>
> Ihr Elastic Cluster muss sich in dem `Available` Zustand befinden, in dem ein manueller Snapshot erstellt werden kann.

In diesem Abschnitt wird erklärt, wie Sie Elastic Cluster-Snapshots erstellen, anzeigen, wiederherstellen und löschen können.

Die folgenden Themen zeigen, wie Sie verschiedene Aufgaben bei der Arbeit mit Amazon DocumentDB Elastic Cluster-Snapshots ausführen.

Themen

- Einen manuellen Elastic Cluster-Snapshot erstellen
- Einen Elastic Cluster-Snapshot anzeigen
- Wiederherstellen eines Elastic Clusters aus einem Snapshot
- Einen Elastic Cluster-Snapshot wird kopiert
- Löschen eines Elastic Cluster-Snapshots
- Verwaltung eines automatischen Backups eines Elastic Cluster-Snapshots

Einen manuellen Elastic Cluster-Snapshot erstellen

In diesem Abschnitt wird anhand der folgenden Anweisungen erklärt, wie Sie einen manuellen Elastic Cluster-Snapshot erstellen. AWS Management Console AWS CLI

Using the AWS Management Console

Um einen manuellen Elastic Cluster-Snapshot zu erstellen, verwenden Sie AWS Management Console:

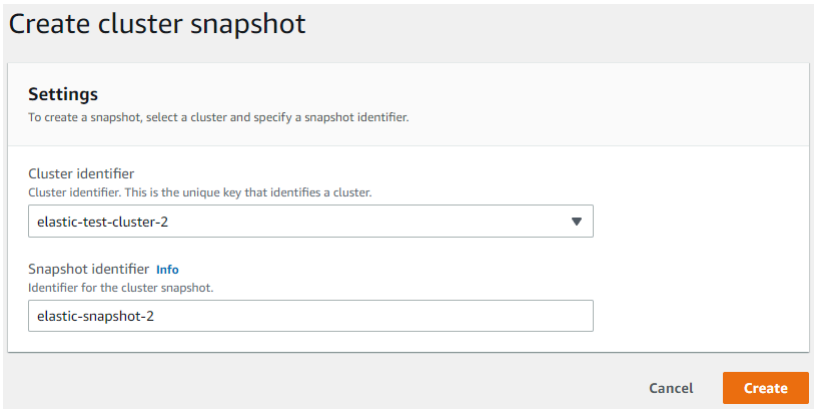
1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   > **Tip**
   >
   > Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie auf der Seite Snapshots die Option Create (Erstellen) aus.
4. Wählen Sie auf der Seite Cluster-Snapshot erstellen im Feld Cluster-ID Ihren Elastic Cluster aus der Dropdownliste aus.

   Geben Sie im Feld Snapshot-Identifier eine eindeutige Kennung für Ihren Elastic Cluster ein.

   Wählen Sie Erstellen aus.



> **Note**
>
> Alternativ können Sie auf das Dialogfeld Cluster-Snapshot erstellen zugreifen, indem Sie auf der Seite Cluster das Kästchen neben Ihrem Cluster markieren und dann Aktionen und dann Snapshot erstellen auswählen.

Ihr Elastic Cluster-Snapshot wird jetzt bereitgestellt. Es kann bis zu ein paar Minuten dauern, bis dieser Vorgang abgeschlossen ist. Sie können Ihren Snapshot ansehen und aus diesem wiederherstellen, wenn der Status wie `Available` in der Snapshot-Liste angezeigt wird.

Using the AWS CLI

Um einen manuellen Elastic Cluster-Snapshot mit dem zu erstellen AWS CLI, verwenden Sie den `create-cluster-snapshot` Vorgang mit den folgenden Parametern:

- **`--snapshot-name`**—Erforderlich. Der Name des Cluster-Snapshots, den Sie erstellen möchten.
- **`--cluster-arn`**—Erforderlich. Die ARN-ID des Clusters, von dem Sie einen Snapshot erstellen möchten.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic create-cluster-snapshot \
    --snapshot-name sample-snapshot-1 \
```

```
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster
```

Für Windows:

```
aws docdb-elastic create-cluster-snapshot ^
    --snapshot-name sample-snapshot-1 ^
    --cluster-arn arn:aws:docdb:us-west-2:123456789012:sharded-cluster:sample-
cluster
```

Einen Elastic Cluster-Snapshot anzeigen

In diesem Abschnitt wird anhand der folgenden Anweisungen erklärt, wie Sie Elastic Cluster-Snapshot-Informationen anzeigen können. AWS Management Console AWS CLI

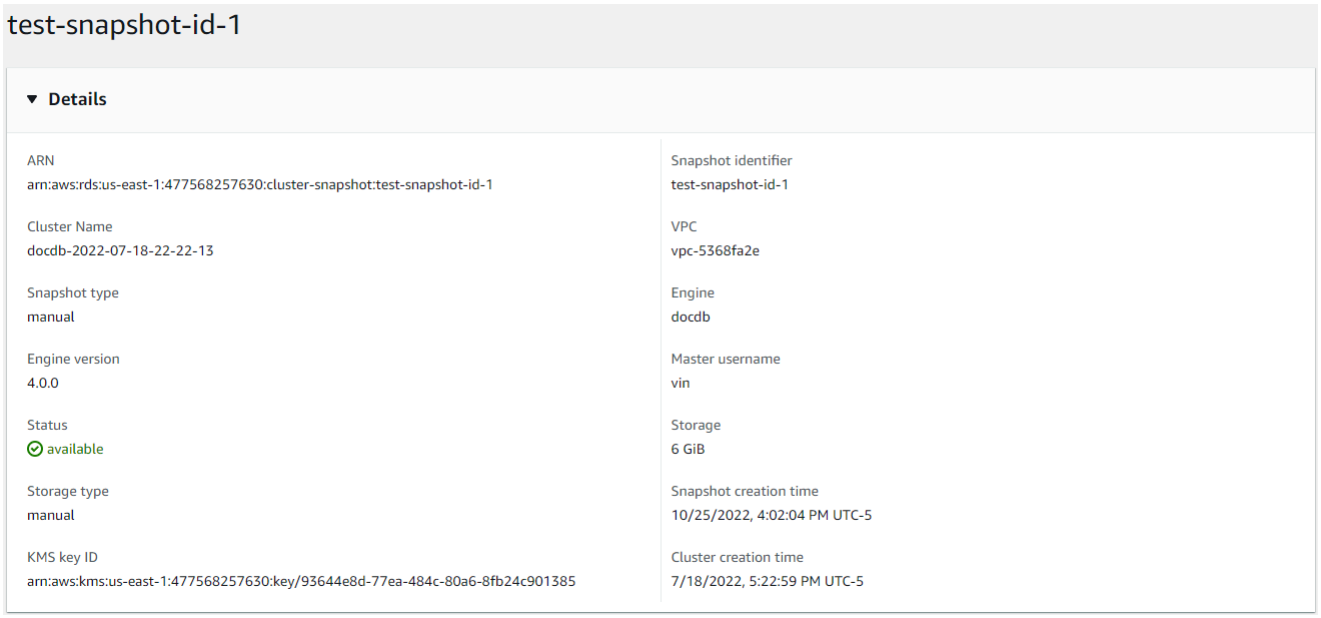
Using the AWS Management Console

Um Informationen zu einem bestimmten Elastic Cluster-Snapshot anzuzeigen, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   > **Tip**
   >
   > Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie auf der Seite Snapshots Ihren Snapshot aus der Liste aus, indem Sie in der Spalte Snapshot-ID auf den Namen klicken.
4. Sehen Sie sich die Informationen zu Ihrem Snapshot unter Details an.



Using the AWS CLI

Verwenden Sie den `get-cluster-snapshot` Vorgang mit den folgenden Parametern AWS CLI, um Informationen zu einem bestimmten Elastic Cluster-Snapshot mithilfe von anzuzeigen:

- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, für den Sie Informationen benötigen.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Für Windows:

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Verwenden Sie den `get-cluster-snapshot` Vorgang mit den folgenden Parametern AWS CLI, um Informationen zu einem bestimmten Elastic Cluster-Snapshot mithilfe von anzuzeigen:

- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, für den Sie Informationen benötigen.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic get-cluster-snapshot \
    --snapshot-arn sampleResourceName
```

Für Windows:

```
aws docdb-elastic get-cluster-snapshot ^
    --snapshot-arn sampleResourceName
```

Verwenden Sie den `list-cluster-snapshots` Vorgang mit den folgenden Parametern, um Informationen zu allen Elastic Cluster-Snapshots anzuzeigen AWS CLI, die den verwenden:

- **`--snapshot-type`**— Optional. Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:
  - `automated`- Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr AWS Konto erstellt hat.
  - `manual`- Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr Konto erstellt haben. AWS
  - `shared`- Gibt alle manuellen Cluster-Snapshots zurück, die für Ihr AWS Konto freigegeben wurden.
  - `public`- Gibt alle Cluster-Snapshots zurück, die als öffentlich markiert wurden.
- **`--next-token`**— Fakultativ. Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn dieser Parameter angegeben ist, umfasst die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von `max-results` angegebenen Wert.
- **`--max-results`**— Optional. Die maximale Anzahl von Ergebnissen, die in die Antwort aufgenommen werden sollen. Wenn mehr Ergebnisse als der angegebene `max-results` Wert

vorhanden sind, ist ein Paginierungstoken (`next-token`) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

- Standard: 100
- Mindestens 20, maximal 100

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic list-cluster-snapshots \
    --snapshot-type  value \
    --next-token  value \
    --max-results 50
```

Für Windows:

```
aws docdb-elastic list-cluster-snapshots ^
    --snapshot-type  value ^
    --next-token  value ^
    --max-results 50
```

Wiederherstellen eines Elastic Clusters aus einem Snapshot

In diesem Abschnitt wird anhand der folgenden Anweisungen erklärt, wie Sie einen Elastic Cluster aus einem Snapshot wiederherstellen, indem Sie entweder die AWS Management Console oder AWS CLI verwenden.

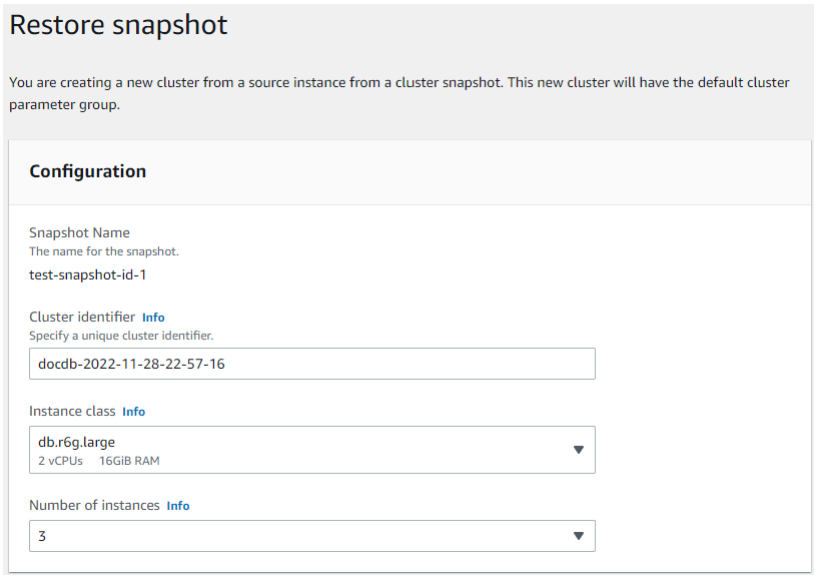
Using the AWS Management Console

Um einen Elastic Cluster aus einem Snapshot wiederherzustellen, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Console und öffnen Sie sie.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

> **Tip**
>
> Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Snapshot-ID die Schaltfläche links neben dem Snapshot, den Sie zum Wiederherstellen eines Clusters verwenden möchten.
4. Wählen Sie „Aktionen“ und dann „Wiederherstellen“.



5. Geben Sie auf der Seite „Snapshot wiederherstellen“ einen Namen für den neuen Cluster in das Feld Cluster-ID ein.

> **Note**
>
> Für jede manuelle Snapshot-Wiederherstellung müssen Sie einen neuen Cluster erstellen.

6. Wählen Sie im Feld Virtual Private Cloud (VPC) eine VPC aus der Drop-down-Liste aus.
7. Für Subnetze und VPC-Sicherheitsgruppen können Sie die Standardeinstellungen verwenden oder drei Subnetze Ihrer Wahl und bis zu drei VPC-Sicherheitsgruppen (mindestens eine) auswählen.
8. Wenn Sie mit der Cluster-Konfiguration zufrieden sind, wählen Sie Restore cluster (Cluster wiederherstellen) aus und warten Sie, bis Ihr Cluster wiederhergestellt ist.

Using the AWS CLI

Verwenden Sie den Vorgang mit den folgenden Parametern, um einen Elastic Cluster aus einem Snapshot mithilfe von wiederherzustellen: AWS CLI`restore-cluster-from-snapshot`

- **`--cluster-name`**—Erforderlich. Der aktuelle Name des elastischen Clusters, wie er bei der Erstellung eingegeben oder zuletzt geändert wurde.
- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, der zur Wiederherstellung des Clusters verwendet wird.
- **`--vpc-security-group-ids`**— Fakultativ. Eine oder mehrere Amazon EC2 - und Amazon Virtual Private Cloud (VPC) -Sicherheitsgruppen, die dem Cluster zugeordnet werden sollen.
- **`--kms-key-id`**— Optional. Konfigurieren Sie die KMS-Schlüssel-ID für einen verschlüsselten Cluster.

  Die KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon Web Services Services-Konto erstellen, dem der KMS-Verschlüsselungsschlüssel gehört, der zur Verschlüsselung des neuen Clusters verwendet wird, können Sie den KMS-Schlüsselalias anstelle des ARN für den KMS-Verschlüsselungsschlüssel verwenden.

  Wenn in kein Verschlüsselungsschlüssel angegeben ist KmsKeyId und der `StorageEncrypted` Parameter wahr ist, verwendet Amazon DocumentDB Ihren Standard-Verschlüsselungsschlüssel.
- **`--subnet-ids`**— Optional. Netzwerk-Subnetz-IDs.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic restore-cluster-from-snapshot \
     --cluster-name elastic-sample-cluster \
     --snapshot-arn sampleResourceName \
     --vpc-security-group-ids value ec-65f40350 \
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 \
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Für Windows:

```
aws docdb-elastic restore-cluster-from-snapshot ^
     --cluster-name elastic-sample-cluster ^
     --snapshot-arn sampleResourceName ^
     --vpc-security-group-ids value ec-65f40350 ^
     --kms-key-id arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2 ^
     --subnet-ids subnet-9253c6a3, subnet-9f1b5af9
```

Einen Elastic Cluster-Snapshot wird kopiert

In Amazon DocumentDB können Sie manuelle und automatische Elastic Cluster-Snapshots innerhalb derselben Region und innerhalb desselben Kontos kopieren. In diesem Abschnitt wird erklärt, wie Sie einen Elastic Cluster-Snapshot kopieren, indem Sie entweder das AWS Management Console oder verwenden. AWS CLI

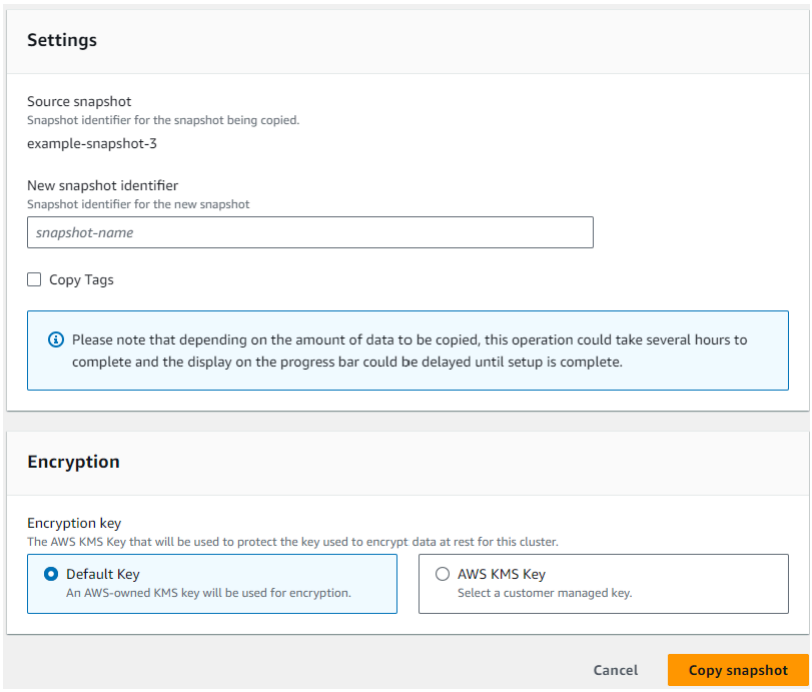
Using the AWS Management Console

Um einen Elastic Cluster-Snapshot zu kopieren, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

   > **Tip**
   >
   > Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Snapshot-ID die Schaltfläche links neben dem Snapshot, den Sie kopieren möchten.
4. Wählen Sie „Aktionen“ und dann „Kopieren“.



5. Geben Sie unter Neuer Snapshot-Identifier den Namen des neuen Snapshots ein.
6. Aktivieren Sie für Copy Tags das Kontrollkästchen, wenn Sie alle Tags aus dem Elastic Cluster-Quell-Snapshot in den Elastic Cluster-Ziel-Snapshot kopieren möchten.
7. Wählen Sie für die Verschlüsselung entweder einen AWS Standard-KMS-Schlüssel oder einen KMS-Schlüssel Ihrer Wahl. Mit der zweiten Option können Sie einen vorhandenen KMS-Schlüssel auswählen, den Sie bereits erstellt haben, oder einen neuen erstellen.
8. Wählen Sie Snapshot kopieren, wenn Sie fertig sind.

Using the AWS CLI

Um einen Elastic Cluster-Snapshot mit dem zu kopieren AWS CLI, verwenden Sie den `copy-cluster-snapshot` Vorgang mit den folgenden Parametern:

- **`--source-db-cluster-snapshot-identifier`**—Erforderlich. Der Bezeichner des vorhandenen Elastic Cluster-Snapshots, der kopiert wird. Der Elastic Cluster-Snapshot muss vorhanden sein und sich im verfügbaren Zustand befinden. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- **`--target-db-cluster-snapshot-identifier`**—Erforderlich. Die ID des neuen Elastic Cluster-Snapshots, der aus dem vorhandenen Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen beim Namen des Ziel-Snapshots:

- Kann nicht der Namen eines vorhandenen Snapshots sein.
- Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche.
- Muss mit einem Buchstaben beginnen.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Ersetzen Sie im folgenden Beispiel jede durch Ihre *user input placeholder* eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic copy-cluster-snapshot \
     --source-cluster-snapshot-arn <sample ARN> \
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Für Windows:

```
aws docdb-elastic copy-cluster-snapshot ^
     --source-cluster-snapshot-arn <sample ARN> ^
     --target-cluster-snapshot-name my-target-copied-snapshot
```

Löschen eines Elastic Cluster-Snapshots

In diesem Abschnitt wird erklärt, wie Sie einen Elastic Cluster-Snapshot löschen, indem Sie entweder das AWS Management Console oder verwenden AWS CLI.

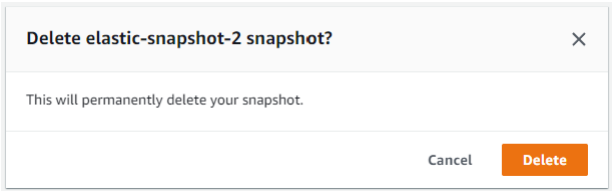
Using the AWS Management Console

Um einen Elastic Cluster aus einem Snapshot wiederherzustellen, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Console und öffnen Sie sie.
2. Wählen Sie im Navigationsbereich die Option Snapshots.

> **Tip**
>
> Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Snapshot-ID die Schaltfläche links neben dem Snapshot, den Sie zum Wiederherstellen eines Clusters verwenden möchten.
4. Wählen Sie Aktionen und dann Löschen aus.



5. Wählen Sie im Dialogfeld „Snapshot-Name“ -Snapshot löschen die Option Löschen aus.

Using the AWS CLI

Um einen Elastic Cluster-Snapshot mit dem zu löschen AWS CLI, verwenden Sie den `delete-cluster-snapshot` Vorgang mit den folgenden Parametern:

- **`--snapshot-arn`**—Erforderlich. Die ARN-ID des Snapshots, der zur Wiederherstellung des Clusters verwendet wird.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Für Linux, macOS oder Unix:

```
aws docdb-elastic delete-cluster-snapshot \
     --snapshot-arn sampleResourceName
```

Für Windows:

```
aws docdb-elastic delete-cluster-snapshot ^
     --snapshot-arn sampleResourceName
```

Verwaltung eines automatischen Backups eines Elastic Cluster-Snapshots

Amazon DocumentDB erstellt täglich Schnappschüsse Ihrer elastischen Cluster. Sie können das bevorzugte Backup-Fenster und den Aufbewahrungszeitraum für Backups in einer neuen oder vorhandenen Elastic Cluster-Snapshot-Konfiguration angeben. In diesem Abschnitt wird erklärt, wie Sie automatische Backup-Parameter in einem Elastic Cluster-Snapshot festlegen, indem Sie entweder das AWS Management Console oder verwenden AWS CLI.

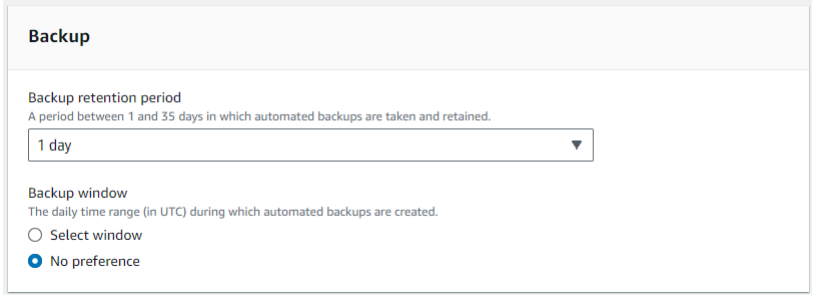
Using the AWS Management Console

Um ein automatisches Backup für einen neuen Elastic Cluster-Snapshot einzurichten, verwenden Sie AWS Management Console:

1. Melden Sie sich bei der Amazon DocumentDB DocumentDB-Konsole an AWS Management Consoleund öffnen Sie sie.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn Sie den Navigationsbereich nicht auf der linken Seite Ihres Bildschirms sehen, wählen Sie das Menüsymbol in der oberen linken Ecke des Navigationsbereichs.

3. Wählen Sie in der Spalte Cluster-ID die Schaltfläche links neben dem Cluster, für den Sie die Backup-Einstellungen ändern möchten.
4. Wählen Sie „Aktionen“ und dann „Ändern“.
5. Bearbeiten Sie im Abschnitt Backup die Felder entsprechend Ihren Backup-Anforderungen.



   a. Aufbewahrungszeitraum für Backup — Wählen Sie in der Liste die Anzahl der Tage aus, für die automatische Backups dieses Clusters aufbewahrt werden sollen, bevor sie gelöscht werden.

b. Backup-Fenster — Legen Sie die tägliche Uhrzeit und Dauer fest, während der Amazon DocumentDB Backups dieses Clusters erstellen soll.

   i. Wählen Sie Fenster auswählen, wenn Sie die Uhrzeit und Dauer der Erstellung von Backups konfigurieren möchten.

      Startzeit — Wählen Sie in der ersten Liste die Startzeit (UTC) für den Start Ihrer automatischen Backups aus. Wählen Sie in der zweiten Liste die Minute für den Beginn der automatischen Backups aus.

      Dauer — Wählen Sie in der Liste die Anzahl der Stunden aus, die für die Erstellung automatischer Backups reserviert werden sollen.

   ii. Wählen Sie Keine Präferenz, wenn Amazon DocumentDB den Zeitpunkt und die Dauer der Erstellung von Backups wählen soll.

6. Wählen Sie Cluster modifizieren, wenn der Vorgang abgeschlossen ist.

Using the AWS CLI

Um ein automatisches Backup für einen neuen Elastic Cluster-Snapshot mithilfe von einzurichten AWS CLI, verwenden Sie den `create-cluster-snapshot` Vorgang mit den folgenden Parametern:

- **`--preferred-backup-window`**— Optional. Der bevorzugte tägliche Zeitraum, in dem automatische Backups erstellt werden. Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils AWS-Region 8 Stunden ausgewählt wird.

  Einschränkungen:

  - Muss im Format `hh24:mi-hh24:mi` angegeben werden.
  - Muss in Universal Coordinated Time (UTC) angegeben werden.
  - Darf nicht mit dem bevorzugten Wartungsfenster in Konflikt treten.
  - Muss mindestens 30 Minuten betragen.
- **`--backup-retention-period`**— Fakultativ. Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Der Standardwert lautet 1.

  Einschränkungen:

  - Es muss ein Mindestwert von 1 angegeben werden.

- Der Bereich liegt zwischen 1 und 35.

> **Note**
>
> Automatisierte Backups werden nur erstellt, wenn sich der Cluster im Status „aktiv“ befindet.

> **Note**
>
> Sie können auch die `backup-retention-period` Parameter `preferred-backup-window` und eines vorhandenen Elastic Clusters mithilfe des `aws docdb-elastic update-cluster` Befehls ändern.

Ersetzen Sie im folgenden Beispiel jede Information *`user input placeholder`* durch Ihre eigenen Informationen.

Im folgenden `create-cluster` Beispiel wird der elastische Amazon DocumentDB-Cluster *`sample-cluster`* mit einer Aufbewahrungsdauer für automatische Backups von *`7`* Tagen und einem bevorzugten Backup-Fenster von *`18:00-18:30 UTC`* erstellt.

Für Linux, macOS oder Unix:

```
aws docdb-elastic create-cluster \
     --cluster-name sample-cluster \
     --shard-capacity 2 \
     --shard-count 2 \
     --admin-user-name SampleAdmin \
     --auth-type PLAIN_TEXT \
     --admin-user-password SamplePass123! \
     --preferred-backup-window 18:00-18:30 \
     --backup-retention-period 7
```

Für Windows:

```
aws docdb-elastic create-cluster ^
     --cluster-name sample-cluster ^
```

```
    --shard-capacity 2 ^
    --shard-count 2 ^
    --admin-user-name SampleAdmin ^
    --auth-type PLAIN_TEXT ^
    --admin-user-password SamplePass123! ^
    --preferred-backup-window 18:00-18:30 ^
    --backup-retention-period 7
```

Stoppen und Starten eines elastischen Amazon DocumentDB-Clusters

Das Stoppen und Starten von Amazon DocumentDB Elastic Clusters kann Ihnen helfen, die Kosten für Entwicklungs- und Testumgebungen zu kontrollieren. Anstatt jedes Mal, wenn Sie Amazon DocumentDB verwenden, elastische Cluster zu erstellen und zu löschen, können Sie Ihren Cluster vorübergehend beenden, wenn er nicht benötigt wird. Sie können ihn dann erneut starten, wenn Sie Ihre Tests fortsetzen.

Themen

- Überblick über das Stoppen und Starten eines Elastic Clusters
- Operationen, die Sie auf einem gestoppten Elastic Cluster ausführen können

Überblick über das Stoppen und Starten eines Elastic Clusters

In Zeiten, in denen Sie keinen elastischen Amazon DocumentDB-Cluster benötigen, können Sie den Cluster beenden. Bei Bedarf können Sie den Cluster jederzeit erneut starten. Das Starten und Stoppen vereinfacht die Einrichtungs- und Abbauprozesse für elastische Cluster, die für Entwicklungs-, Test- oder ähnliche Aktivitäten verwendet werden, für die keine kontinuierliche Verfügbarkeit erforderlich ist. Sie können einen Elastic Cluster mithilfe von AWS Management Console oder AWS CLI mit einer einzigen Aktion beenden und starten.

Während Ihr Elastic Cluster gestoppt ist, bleibt das Cluster-Speichervolumen unverändert. Sie zahlen nur für Speicherung, manuelle Snapshots und automatischen Sicherungsspeicher innerhalb des angegebenen Aufbewahrungsfensters. Amazon DocumentDB startet Ihren Elastic Cluster automatisch nach sieben Tagen, sodass er nicht hinter den erforderlichen Wartungsupdates zurückbleibt. Wenn Ihr Cluster nach sieben Tagen startet, wird Ihnen die Nutzung des Elastic Clusters wieder in Rechnung gestellt. Solange Ihr Cluster gestoppt ist, können Sie Ihr Speichervolume nicht abfragen, da die Abfrage voraussetzt, dass sich der Cluster im verfügbaren Status befindet.

Wenn ein elastischer Amazon DocumentDB-Cluster gestoppt wird, kann der Cluster in keiner Weise geändert werden. Dies beinhaltet das Löschen des Clusters.

Using the AWS Management Console

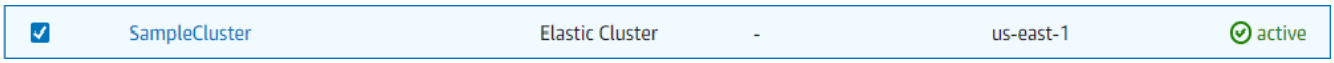
Das folgende Verfahren zeigt Ihnen, wie Sie einen elastischen Cluster im verfügbaren Zustand beenden oder einen gestoppten elastischen Cluster starten.

So beenden oder starten Sie einen elastischen Amazon DocumentDB-Cluster

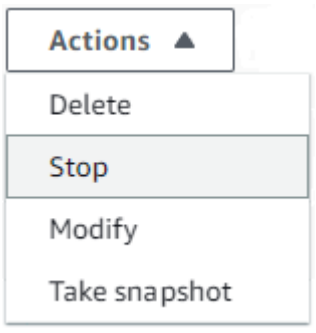
1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Tip**
>
> Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

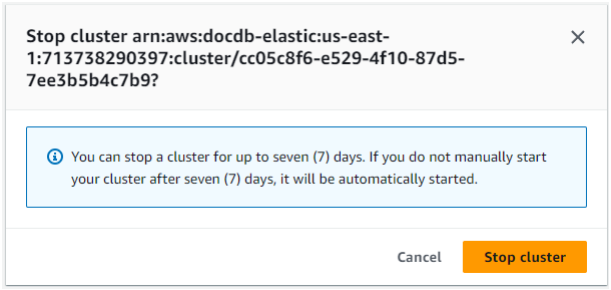
3. Wählen Sie in der Liste der Cluster die Schaltfläche links neben dem Namen des Clusters aus, den Sie stoppen oder starten möchten.



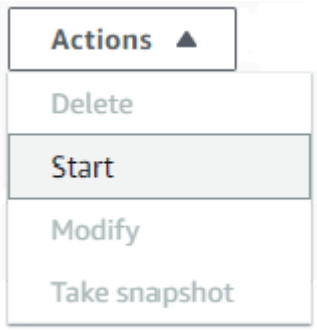
4. Wählen Sie Aktionen aus und dann die Aktion, die Sie auf dem Cluster ausführen möchten.
   - Wenn Sie den Cluster stoppen möchten und der Cluster verfügbar ist:
     a. Wählen Sie Beenden aus.



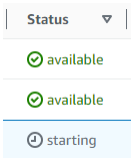
b. Bestätigen Sie im Bestätigungsdialogfeld, dass Sie den Elastic Cluster beenden möchten, indem Sie Cluster beenden wählen, oder klicken Sie auf Abbrechen, um den Cluster am Laufen zu halten.



- Wenn Sie einen gestoppten Cluster starten möchten, wählen Sie Starten aus.



5. Überwachen Sie den Status des Elastic Clusters. Wenn Sie den Cluster gestartet haben, können Sie den Cluster wieder verwenden, sobald der Cluster verfügbar ist. Weitere Informationen finden Sie unter Den Status eines Clusters ermitteln.



Using the AWS CLI

Die folgenden Codebeispiele zeigen Ihnen, wie Sie einen elastischen Cluster im aktiven oder verfügbaren Zustand beenden oder einen gestoppten elastischen Cluster starten.

Verwenden Sie die `stop-cluster` Operation AWS CLI, um einen elastischen Cluster mithilfe von zu beenden. Verwenden Sie die `start-cluster`-Operation, um einen gestoppten Cluster zu starten. Beide Operationen verwenden den `--cluster-arn`-Parameter.

Parameter:

- **`--cluster-arn`**—Erforderlich. Der ARN-Bezeichner des Elastic Clusters, den Sie beenden oder starten möchten.

Example — Um einen elastischen Cluster mit dem zu beenden AWS CLI

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Der folgende Code stoppt den elastischen Cluster mit einem ARN von`arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`.

> **Note**
>
> Der elastische Cluster muss sich im aktiven oder verfügbaren Zustand befinden.

Für Linux, macOS oder Unix:

```
aws docdb-elastic stop-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Für Windows:

```
aws docdb-elastic stop-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Example — Um einen elastischen Cluster mit dem zu starten AWS CLI

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch Ihre eigenen Informationen.

Der folgende Code startet den Elastic Cluster mit einem ARN von`arn:aws:docdb-elastic:us-east-1:477568257630:cluster/b9f1d489-6c3e-4764-bb42-da62ceb7bda2`.

> **Note**
>
> Der elastische Cluster muss derzeit gestoppt werden.

Für Linux, macOS oder Unix:

```
aws docdb-elastic start-cluster \
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Für Windows:

```
aws docdb-elastic start-cluster ^
   --cluster-arn arn:aws:docdb-elastic:us-east-1:477568257630:cluster/
b9f1d489-6c3e-4764-bb42-da62ceb7bda2
```

Operationen, die Sie auf einem gestoppten Elastic Cluster ausführen können

Sie können die Konfiguration eines elastischen Amazon DocumentDB-Clusters nicht ändern, solange der Cluster gestoppt ist. Vor solchen administrativen Aktionen müssen Sie den Cluster starten.

Amazon DocumentDB wendet alle geplanten Wartungsarbeiten auf Ihren gestoppten Elastic Cluster erst an, nachdem er erneut gestartet wurde. Nach sieben Tagen startet Amazon DocumentDB automatisch einen gestoppten Elastic Cluster, sodass er in seinem Wartungsstatus nicht zu weit zurückfällt. Wenn der Elastic Cluster neu gestartet wird, werden Ihnen die Shards im Cluster wieder in Rechnung gestellt.

Während ein elastischer Cluster gestoppt ist, führt Amazon DocumentDB keine automatisierten Backups durch und verlängert auch nicht den Aufbewahrungszeitraum für Backups.

Wartung elastischer Amazon DocumentDB-Cluster

Themen

- Ausstehende Wartungsaktionen für Elastic Cluster anzeigen
- Aktualisierungen der Elastic Cluster Engine
- Betriebssystem-Updates für Elastic Cluster

Amazon DocumentDB führt regelmäßig Wartungsarbeiten an den Elastic Cluster-Ressourcen von Amazon DocumentDB durch. Die Wartung umfasst in den meisten Fällen Aktualisierungen der Datenbank-Engine (Elastic Cluster-Wartung) oder des dem Elastic Cluster zugrunde liegenden Betriebssystems (Betriebssystem-Updates). Datenbank-Engine-Updates sind erforderliche Patches und beinhalten Sicherheitskorrekturen, Bugfixes und Verbesserungen der Datenbank-Engine.
Die meisten Betriebssystem-Patches sind zwar optional, aber wenn Sie sie eine Zeit lang nicht anwenden, kann es sein, dass der Patch erforderlich ist und auto angewendet wird, um Ihren Sicherheitsstatus aufrechtzuerhalten. Wir empfehlen Ihnen daher, Betriebssystem-Updates auf Ihre elastischen Amazon DocumentDB-Cluster zu installieren, sobald sie verfügbar sind.

Datenbank-Engine-Patches erfordern, dass Sie Ihre elastischen Amazon DocumentDB-Cluster für kurze Zeit offline schalten. Sobald sie verfügbar sind, werden diese Patches automatisch so geplant, dass sie während eines bevorstehenden geplanten Wartungsfensters Ihres Amazon DocumentDB Elastic Clusters installiert werden.

Elastische Cluster haben ihre eigenen jeweiligen Wartungsfenster. Elastic Cluster-Änderungen, die Sie nicht sofort anwenden möchten, werden während des Wartungsfensters angewendet. Wenn Sie einen Elastic Cluster erstellen, weist Amazon DocumentDB standardmäßig ein Wartungsfenster für Ihren Elastic Cluster zu. Sie können das Wartungsfenster wählen, wenn Sie einen elastischen Cluster erstellen. Sie können die Wartungsfenster außerdem jederzeit an Ihre Geschäftspläne oder -praktiken anpassen. Es wird generell empfohlen, Wartungsfenster auszuwählen, die die Auswirkungen der Wartung auf Ihre Anwendung minimieren (z. B. an Abenden oder Wochenenden).

Ausstehende Wartungsaktionen für Elastic Cluster anzeigen

Mithilfe von können Sie überprüfen, ob ein Wartungsupdate für Ihren Elastic Cluster verfügbar ist AWS CLI.

Wenn ein Update verfügbar ist, können Sie eine der folgenden Aktionen durchführen:

- Verschieben Sie eine Wartungsaktion, die derzeit für das nächste Wartungsfenster geplant ist (nur für Betriebssystem-Patches).
- Sofortiges Durchführen der Wartungsaktivitäten-
- Einplanen der Wartungsaktivitäten für das nächste Wartungsfenster.
- Planen Sie die Wartungsaktionen so, dass sie während des ausgewählten Anwendungsfensters beginnen.

Das Wartungsfenster legt fest, wann die ausstehenden Operationen gestartet werden, gibt aber kein Abschlussdatum für diese Operationen vor; die Ausführungsdauer ist weder bekannt noch a priori beschränkt.

Verwenden Sie den folgenden AWS CLI Vorgang, um festzustellen, welche Wartungsaktionen noch ausstehen. Listet alle ausstehenden Wartungsaktionen auf:

```
aws docdb-elastic list-pending-maintenance-actions
```

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
{
'ResourcePendingMaintenanceActions': [
    {
        'ResourceArn': 'string-arn',
        'PendingMaintenanceActionDetails': [
            {
                'Action': 'ENGINE_UPDATE',
                'AutoAppliedAfterDate': 'string',
                'ForcedApplyDate': 'string',
                'OptInStatus': 'string',
                'CurrentApplyDate': 'string',
                'Description': 'string'
            },
        ]
    },
],
'NextToken': 'string'
}
```

Ruft ausstehende Wartungsmaßnahmen (falls vorhanden) für ein bestimmtes Objekt ab`resourceArn`:

```
aws docdb-elastic get-pending-maintenance-action --resource-arn string-arn
```

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

```
{
    'ResourcePendingMaintenanceAction': {
        'ResourceArn': 'string-arn',
        'PendingMaintenanceActionDetails': [
            {
```

```
                'Action': 'ENGINE_UPDATE',
                'AutoAppliedAfterDate': 'string',
                'ForcedApplyDate': 'string',
                'OptInStatus': 'string',
                'CurrentApplyDate': 'string',
                'Description': 'string'
            }
        ]
    }
}
```

Parameter:

- `ResourceArn`— Der Amazon Resource Name (ARN) von Amazon DocumentDB der Ressource, für die die ausstehende Wartungsaktion gilt.
- `Action`— Die ausstehende Wartungsaktion, die auf die Ressource angewendet wird.

  Zulässige Werte:

  - `ENGINE_UPDATE`
  - `ENGINE_UPGRADE`
  - `SECURITY_UPDATE`
  - `OS_UPDATE`
  - `MASTER_USER_PASSWORD_UPDATE`
- `AutoAppliedAfterDate`— Erstes Wartungsfenster nach diesem Datum. `NEXT_MAINTENANCE` `OPT_IN`wird in diesem Fall ignoriert.
- `ForcedApplyDate`— Wird unabhängig vom Wartungsfenster angewendet. `IMMEDIATE` `OPT_IN`wird in diesem Fall ignoriert.
- `OptInStatus`— Ein Wert, der den Typ der Opt-in-Anfrage angibt oder eine Opt-in-Anfrage rückgängig macht. Eine Opt-in-Anfrage vom Typ `IMMEDIATE` kann nicht rückgängig gemacht werden.

  Zulässige Werte:

  - `IMMEDIATE`— Wendet die Wartungsmaßnahme sofort an.
  - `NEXT_MAINTENANCE`— Wendet die Wartungsaktion im nächsten Wartungsfenster für die Ressource an.
  - `APPLY_ON`— Wendet die Wartungsaktion unabhängig vom nächsten Wartungsfenster für die Ressource am angegebenen Anwendungsdatum an.

  - `UNDO_OPT_IN`— Storniert alle bestehenden Anfragen `NEXT_MAINTENANCE` oder `APPLY_ON` Opt-In-Anfragen.
- `CurrentApplyDate`— Wird angezeigt, falls ja opt-in-type. `APPLY_ON`
- `Description`— Eine Beschreibung der Option für die Wartungsaktion.

Aktualisierungen der Elastic Cluster Engine

Mit Amazon DocumentDB können Sie wählen, wann Wartungsarbeiten durchgeführt werden sollen. Mit dem können Sie entscheiden, wann Amazon DocumentDB Updates AWS CLI einführt.

Ausstehende Wartungsmaßnahmen anwenden:

```
aws docdb-elastic apply-pending-maintenance-action
--resource-arn string-arn
--apply-action string-enum
--opt-in-type string-enum
[--apply-on string-date-range]
```

Parameter:

- **`--resource-arn`**— Der Amazon Resource Name (ARN) von Amazon DocumentDB der Ressource, für die die ausstehende Wartungsaktion gilt.
- **`--apply-action`**— Die ausstehende Wartungsaktion, die auf diese Ressource angewendet werden soll.

  Zulässige Werte:

  - `ENGINE_UPDATE`
  - `ENGINE_UPGRADE`
  - `SECURITY_UPDATE`
  - `OS_UPDATE`
  - `MASTER_USER_PASSWORD_UPDATE`
- **`--opt-in-type`**— Ein Wert, der den Typ der Opt-in-Anfrage angibt oder eine Opt-in-Anfrage rückgängig macht. Eine Opt-in-Anfrage vom Typ `IMMEDIATE` kann nicht rückgängig gemacht werden.

  Zulässige Werte:

  - `IMMEDIATE`— Wendet die Wartungsmaßnahme sofort an.

- `NEXT_MAINTENANCE`— Wendet die Wartungsaktion im nächsten Wartungsfenster für die Ressource an.
- `APPLY_ON`— Wendet die Wartungsaktion unabhängig vom nächsten Wartungsfenster für die Ressource am angegebenen Anwendungsdatum an.
- `UNDO_OPT_IN`— Storniert alle bestehenden Anfragen `NEXT_MAINTENANCE` oder `APPLY_ON` Opt-In-Anfragen.

- **`[--apply-on]`**— Erforderlich, falls ja opt-in-type. `APPLY_ON` Format: `yyyy/MM/dd HH:mm-yyyy/MM/dd HH:mm` (Diese Option verwendet UTC-Zeit. Die Startzeit kann ein beliebiger Zeitpunkt in der future sein, zwischen mindestens 30 Minuten und maximal 14 Tagen, oder das Force/Application-Datum für ausstehende Maßnahmen sein, je nachdem, welcher Zeitpunkt früher liegt. Das Zeitfenster von Anfang bis Ende kann mindestens 30 Minuten bis maximal 8 Stunden lang sein.)

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
{
 'ResourcePendingMaintenanceAction': {
        'ResourceArn': 'string-arn',
        'PendingMaintenanceActionDetails': [
            {
                'Action': 'SECURITY_UPDATE',
                'AutoAppliedAfterDate': 'string',
                'ForcedApplyDate': 'string',
                'OptInStatus': 'IMMEDIATE',
                'CurrentApplyDate': 'string',
                'Description': 'string'
            },
        ]
 }
}
```

Parameter:

- `ResourceArn`— Der Amazon Resource Name (ARN) von Amazon DocumentDB der Ressource, für die die ausstehende Wartungsaktion gilt.
- `Action`— Die ausstehende Wartungsaktion, die auf die Ressource angewendet wird.

  Zulässige Werte:

  - ENGINE_UPDATE
  - ENGINE_UPGRADE
  - SECURITY_UPDATE
  - OS_UPDATE
  - MASTER_USER_PASSWORD_UPDATE
- AutoAppliedAfterDate— Erstes Wartungsfenster nach diesem Datum. NEXT_MAINTENANCE OPT_INwird in diesem Fall ignoriert.
- ForcedApplyDate— Wird unabhängig vom Wartungsfenster angewendet. IMMEDIATE OPT_INwird in diesem Fall ignoriert.
- OptInStatus— Ein Wert, der den Typ der Opt-in-Anfrage angibt oder eine Opt-in-Anfrage rückgängig macht. Eine Opt-in-Anfrage vom Typ IMMEDIATE kann nicht rückgängig gemacht werden.

  Zulässige Werte:

  - IMMEDIATE— Wendet die Wartungsmaßnahme sofort an.
  - NEXT_MAINTENANCE— Wendet die Wartungsaktion im nächsten Wartungsfenster für die Ressource an.
  - APPLY_ON— Wendet die Wartungsaktion unabhängig vom nächsten Wartungsfenster für die Ressource am angegebenen Anwendungsdatum an.
  - UNDO_OPT_IN— Storniert alle bestehenden Anfragen NEXT_MAINTENANCE oder APPLY_ON Opt-In-Anfragen.
- CurrentApplyDate— Wird angezeigt, falls ja opt-in-type. APPLY_ON
- Description— Eine Beschreibung der Option für die Wartungsaktion.

Termine anwenden

Jede Wartungsaktivität hat ein entsprechendes Anwendungsdatum, das Sie bei der Beschreibung der anstehenden Wartungsaktivität finden. Wenn Sie die Ausgabe der ausstehenden Wartungsmaßnahmen vom lesen AWS CLI, werden drei Daten aufgeführt:

- CurrentApplyDate— Das Datum, an dem die Wartungsaktion entweder sofort oder während des nächsten Wartungsfensters ausgeführt wird. Wenn die Wartung optional ist, kann dieser Wert Null sein.

- `ForcedApplyDate`— Das Datum, an dem die Wartung unabhängig von Ihrem Wartungsfenster automatisch angewendet wird.
- `AutoAppliedAfterDate`— Das Datum, nach dem die Wartung während des Wartungsfensters des Clusters angewendet wird.

Vom Benutzer erstellte Wartungsaktionen

Als Amazon DBelastic Document-Cluster-Benutzer können Sie Aktualisierungen Ihrer Cluster-Konfigurationen veranlassen.

Das primäre Cluster-Passwort wird aktualisiert

```
aws docdb-elastic update-cluster
--cluster-arn string-arn
[--admin-user-password string]
[--auth-type string-enum]
[--apply-method string-enum]
[--apply-on string-date-range]
#... other parameters of the API that follow here are not relevant for this
 configuration
```

Parameter:

- **`--cluster-arn`**— Der Amazon Resource Name (ARN) von Amazon DocumentDB der Ressource, auf die die Wartungsmaßnahme angewendet wird.
- **`[--admin-user-password]`**— Das dem Admin-Benutzer zugeordnete Passwort.
- **`[--auth-type]`**— Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das für den Zugriff auf den Elastic Cluster verwendete Passwort abgerufen werden soll. Gültige Typen sind `PLAIN_TEXT` oder`SECRET_ARN`.
- **`[--apply-method]`**— Ein Wert, der den Typ der Methode angibt, die angewendet wird. Zulässige Werte sind `IMMEDIATE` und `APPLY_ON`. Der Standardwert ist `IMMEDIATE`.
- **`[--apply-on]`**— Erforderlich, wenn ja`apply-method`. `APPLY_ON` Format: `yyyy/MM/dd HH:mm-yyyy/MM/dd HH:mm` (Diese Option verwendet UTC-Zeit. Die Startzeit kann zu einem beliebigen Zeitpunkt in der future liegen und zwischen mindestens 30 Minuten und maximal 14 Tagen liegen. Das Zeitfenster von Anfang bis Ende kann mindestens 30 Minuten bis maximal 8 Stunden lang sein.)

Die Ausgabe dieses Vorgangs sieht etwa wie folgt aus (JSON-Format):

```
{
 'ResourcePendingMaintenanceAction': {
        'ResourceArn': 'string-arn',
        'PendingMaintenanceActionDetails': [
            {
                'Action': 'MASTER_USER_PASSWORD_UPDATE',
                'OptInStatus': 'APPLY_ON',
                'CurrentApplyDate': 'string',
                'Description': 'string'
            },
        ]
 }
}
```

Ihre Amazon DocumentDB DocumentDB-Wartungsfenster ändern

Das Wartungsfenster sollte in den Zeitraum mit der geringsten Nutzung fallen und daher unter Umständen von Zeit zu Zeit geändert werden. Ihr Elastic Cluster ist während dieser Zeit nur dann nicht verfügbar, wenn Systemänderungen (z. B. eine Änderung des Speicherbetriebs) vorgenommen werden und ein Ausfall erforderlich ist. Er ist nur für den Mindestzeitraum nicht verfügbar, der erforderlich ist, um die erforderlichen Änderungen vorzunehmen.

Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock für jede Amazon Web Services Services-Region ausgewählt wird und an einem zufälligen Wochentag stattfindet.

Informationen zum Ändern des Wartungsfensters finden Sie unterModifizieren elastischer Cluster-Konfigurationen.

Betriebssystem-Updates für Elastic Cluster

Elastische Amazon DocumentDB-Cluster erfordern gelegentlich Betriebssystemupdates. Amazon DocumentDB aktualisiert das Betriebssystem auf eine neuere Version, um die Datenbankleistung und die allgemeine Sicherheitslage der Kunden zu verbessern. Betriebssystemupdates ändern nicht die Cluster-Engine-Version eines elastischen Amazon DocumentDB-Clusters.

Die meisten Betriebssystemupdates für Amazon DocumentDB Elastic Clusters sind optional und haben kein festes Datum, an dem sie angewendet werden sollen. Wenn Sie diese Updates jedoch eine Zeit lang nicht installieren, werden sie möglicherweise irgendwann erforderlich und werden

während des Wartungsfensters Ihres Clusters automatisch angewendet. Dies soll dazu beitragen, den Sicherheitsstatus Ihrer Datenbank aufrechtzuerhalten. Um unerwartete Ausfallzeiten zu vermeiden, empfehlen wir Ihnen, Betriebssystem-Updates auf Ihren Amazon DocumentDB Elastic Cluster zu installieren, sobald sie verfügbar sind, und Ihr Cluster-Wartungsfenster zu einem Zeitpunkt festzulegen, der Ihren Geschäftsanforderungen entspricht.

Datenverschlüsselung im Ruhezustand für elastische Amazon DocumentDB-Cluster

In den folgenden Themen erfahren Sie mehr über AWS Key Management Service Verschlüsselungsschlüssel für elastische Amazon DocumentDB-Cluster, deren Erstellung und Überwachung:

Themen

- So verwenden elastische Amazon DocumentDB-Cluster Zuschüsse in AWS KMS
- Einen kundenverwalteten Schlüssel erstellen
- Überwachung Ihrer Verschlüsselungsschlüssel für Amazon DocumentDB Elastic Clusters
- Weitere Informationen

Elastische Amazon DocumentDB-Cluster integrieren sich automatisch in AWS Key Management Service (AWS KMS) für die Schlüsselverwaltung und verwenden eine Methode, die als Envelope-Verschlüsselung bekannt ist, um Ihre Daten zu schützen. Weitere Informationen zur Envelope-Verschlüsselung finden Sie unter Envelope-Verschlüsselung im AWS Key Management Service - Entwicklerhandbuch.

An AWS KMS key ist eine logische Darstellung eines Schlüssels. Der KMS-Schlüssel enthält Metadaten wie die Schlüssel-ID, das Erstellungsdatum, die Beschreibung und den Schlüsselstatus. Der KMS-Schlüssel enthält auch das zur Ver- und Entschlüsselung von Daten verwendete Schlüsselmaterial. Weitere Informationen über KMS-Schlüssel finden Sie unter AWS KMS keys im AWS Key Management Service Developer Guide.

Elastic Clusters von Amazon DocumentDB unterstützen die Verschlüsselung mit zwei Arten von Schlüsseln:

- AWS eigene Schlüssel — Elastische Amazon DocumentDB-Cluster verwenden diese Schlüssel standardmäßig, um personenbezogene Daten automatisch zu verschlüsseln. Sie können AWS

eigene Schlüssel nicht einsehen, verwalten oder verwenden oder deren Verwendung überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme zum Schutz der Schlüssel ändern, die zur Verschlüsselung Ihrer Daten verwendet werden. Weitere Informationen finden Sie unter AWS -eigene Schlüssel im AWS Key Management Service -Entwicklerhandbuch.

- Vom Kunden verwaltete Schlüssel — symmetrische Schlüssel AWS KMS keys , die Sie erstellen, besitzen und verwalten. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:
  - Festlegung und Pflege wichtiger Richtlinien
  - Festlegung und Aufrechterhaltung von IAM-Richtlinien und -Zuschüssen
  - Aktivieren und Deaktivieren wichtiger Richtlinien
  - Kryptographisches Material mit rotierendem Schlüssel
  - Hinzufügen von Tags
  - Erstellen von Schlüsselaliasen
  - Schlüssel für das Löschen von Schlüsseln planen

  Weitere Informationen finden Sie unter Kundenverwaltete Schlüssel im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie müssen einen KMS-Schlüssel für die symmetrische Verschlüsselung verwenden, um Ihren Cluster zu verschlüsseln, da Amazon DocumentDB nur KMS-Schlüssel mit symmetrischer Verschlüsselung unterstützt. Verwenden Sie keinen asymmetrischen KMS-Schlüssel, um zu versuchen, die Daten in Ihren elastischen Amazon DocumentDB-Clustern zu verschlüsseln. Weitere Informationen finden Sie unter Asymmetrische Schlüssel AWS KMS im Entwicklerhandbuch.AWS Key Management Service
> Wenn Amazon DocumentDB keinen Zugriff mehr auf den Verschlüsselungsschlüssel für einen Cluster erhält — zum Beispiel, wenn der Zugriff auf einen Schlüssel gesperrt wird — geht der verschlüsselte Cluster in einen Terminalstatus über. In diesem Fall können Sie den Cluster nur aus einer Sicherung wiederherstellen. Für Amazon DocumentDB sind Backups immer für einen Tag aktiviert. Wenn Sie den Schlüssel für einen verschlüsselten Amazon DocumentDB-Cluster deaktivieren, verlieren Sie außerdem irgendwann den Lese- und Schreibzugriff auf diesen Cluster. Wenn Amazon DocumentDB auf einen Cluster trifft, der mit einem Schlüssel verschlüsselt ist, auf den es keinen Zugriff hat, versetzt es den Cluster in einen Terminalstatus. In diesem Fall ist der Cluster nicht länger verfügbar und der

aktuelle Zustand der Datenbank kann nicht mehr wiederhergestellt werden. Um den Cluster wiederherzustellen, müssen Sie den Zugriff auf den Verschlüsselungsschlüssel für Amazon DocumentDB erneut aktivieren und dann den Cluster aus einem Backup wiederherstellen.

Important

Sie können den KMS-Schlüssel für einen verschlüsselten Cluster nicht ändern, nachdem Sie ihn bereits erstellt haben. Stellen Sie sicher, dass Sie Ihre Anforderungen an den Verschlüsselungsschlüssel ermitteln, bevor Sie Ihren verschlüsselten Elastic Cluster erstellen.

So verwenden elastische Amazon DocumentDB-Cluster Zuschüsse in AWS KMS

Elastische Amazon DocumentDB-Cluster benötigen einen Zuschuss, um Ihren vom Kunden verwalteten Schlüssel verwenden zu können.

Wenn Sie einen Cluster erstellen, der mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, erstellen Amazon DocumentDB Elastic Clusters in Ihrem Namen einen Zuschuss, indem sie eine `CreateGrant` Anfrage an AWS KMS senden. Grants in AWS KMS werden verwendet, um Amazon DocumentDB Elastic Clusters Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

Elastische Amazon DocumentDB-Cluster benötigen den Zuschuss, um Ihren vom Kunden verwalteten Schlüssel für die folgenden internen Operationen verwenden zu können:

- Senden Sie `DescribeKey` Anfragen, um AWS KMS zu überprüfen, ob die symmetrische, vom Kunden verwaltete KMS-Schlüssel-ID, die Sie bei der Erstellung einer Tracker- oder Geofence-Sammlung eingegeben haben, gültig ist.
- Senden Sie `GenerateDataKey` Anfragen AWS KMS zur Generierung von Datenschlüsseln, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden.
- Senden Sie `Decrypt` Anfragen AWS KMS zur Entschlüsselung der verschlüsselten Datenschlüssel, damit diese zur Verschlüsselung Ihrer Daten verwendet werden können.
- Sie können den Zugriff auf die Genehmigung jederzeit widerrufen oder den Zugriff des Services auf den vom Kunden verwalteten Schlüssel entfernen. Wenn Sie dies tun, können Amazon

DocumentDB Elastic Clusters auf keine der mit dem vom Kunden verwalteten Schlüssel verschlüsselten Daten zugreifen, was sich auf Operationen auswirkt, die von diesen Daten abhängig sind.

Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS Management Console oder der AWS KMS API erstellen.

Symmetrische, vom Kunden verwaltete Schlüsselerstellung

Folgen Sie den Schritten zum Erstellen eines symmetrischen kundenverwalteten Schlüssels im Entwicklerhandbuch zum AWS Key Management Service .

Schlüsselrichtlinie

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie in den Informationen zum Zugriff auf KMS-Schlüssel in der AWS Key Management Service Übersicht im AWS Key Management Service Developer Guide.

Um Ihren vom Kunden verwalteten Schlüssel mit Amazon DocumentDB Elastic Cluster-Ressourcen zu verwenden, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zugelassen sein:

- `kms:CreateGrant`: Fügt einem kundenverwalteten Schlüssel eine Erteilung hinzu. Gewährt Kontrollzugriff auf einen bestimmten KMS-Schlüssel, der Zugriff auf Grant-Operationen ermöglicht, die Amazon Location Service benötigt. Weitere Informationen zur Verwendung von Zuschüssen finden Sie unter Grants AWS KMS im AWS Key Management Service Developer Guide.
- `kms:DescribeKey`— Stellt dem Kunden verwaltete Schlüsseldetails zur Verfügung, damit Docdb Elastic den Schlüssel validieren kann.
- `kms:Decrypt`— Ermöglicht Docdb Elastic, den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf verschlüsselte Daten zu verwenden.
- `kms:GenerateDataKey`— Ermöglicht Docdb Elastic, einen verschlüsselten Datenschlüssel zu generieren und ihn zu speichern, da der Datenschlüssel nicht sofort zum Verschlüsseln verwendet wird.

Weitere Informationen finden Sie unter Berechtigungen für AWS Dienste in den wichtigsten Richtlinien und unter Fehlerbehebung beim Schlüsselzugriff im AWS Key Management Service Entwicklerhandbuch.

Einschränkung des vom Kunden verwalteten Schlüsselzugriffs über IAM-Richtlinien

Zusätzlich zu den KMS-Schlüsselrichtlinien können Sie in einer IAM-Richtlinie auch die KMS-Schlüsselberechtigungen einschränken.

Sie können die IAM-Richtlinie auf verschiedene Weise strikter gestalten. Um beispielsweise zu ermöglichen, dass der vom Kunden verwaltete Schlüssel nur für Anfragen verwendet wird, die ihren Ursprung in elastischen Amazon DocumentDB-Clustern haben, können Sie den `kms:ViaServiceBedingungsschlüssel` mit dem `docdb-elastic.<region-name>.amazonaws.com` Wert verwenden.

Weitere Informationen finden Sie unter Benutzern in anderen Konten die Verwendung eines KMS-Schlüssels erlauben im AWS Key Management Service -Entwicklerhandbuch.

Überwachung Ihrer Verschlüsselungsschlüssel für Amazon DocumentDB Elastic Clusters

Wenn Sie einen vom AWS KMS key Kunden verwalteten Schlüssel mit Ihren Docdb Elastic-Ressourcen verwenden, können Sie AWS CloudTrail oder Amazon CloudWatch Logs verwenden, um Anfragen zu verfolgen, an die Docdb Elastic sendet. AWS KMS

Die folgenden Beispiele sind AWS CloudTrail Ereignisse für`CreateGrant`,, und `DescribeKey` zur Überwachung von AWS KMS key Vorgängen `GenerateDataKeyWithoutPlainTextDecrypt`, die von Amazon DocumentDB DocumentDB-Elastic-Clustern aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

CreateGrant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
```

```
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-09T23:04:20Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-09T23:55:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "retiringPrincipal": "docdb-elastic.us-east-1.amazonaws.com",
        "granteePrincipal": "docdb-elastic.us-east-1.amazonaws.com",
        "operations": [
            "Decrypt",
            "Encrypt",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "CreateGrant",
            "RetireGrant",
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
```

```
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateDataKey

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-10T18:02:59Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
```

```
    "eventTime": "2023-05-10T18:03:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
```

```
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-10T18:05:49Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-10T18:06:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


DescribeKey


```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-05-09T23:04:20Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "docdb-elastic.amazonaws.com"
    },
    "eventTime": "2023-05-09T23:55:48Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "docdb-elastic.amazonaws.com",
    "userAgent": "docdb-elastic.amazonaws.com",
    "requestParameters": {
        "keyId": "alias/SampleKmsKey"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
```

```
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Weitere Informationen

Die folgenden Ressourcen bieten weitere Informationen zur Datenverschlüsselung im Ruhezustand:

- Weitere Informationen zu AWS KMS Konzepten finden Sie unter AWS Key Management Service Grundbegriffe im AWS Key Management Service Entwicklerhandbuch.
- Weitere Informationen zur AWS KMS Sicherheit finden Sie unter Bewährte Sicherheitsmethoden für AWS Key Management Service im AWS Key Management Service Entwicklerhandbuch.

Serviceverknüpfte Rollen in elastischen Clustern

Elastische Amazon DocumentDB-Cluster verwenden AWS Identity and Access Management (IAM) service-verknüpfte Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit elastischen Amazon DocumentDB-Clustern verknüpft ist. Servicebezogene Rollen sind von Amazon DocumentDB Elastic Clusters vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Eine serviceverknüpfte Rolle erleichtert die Verwendung von Amazon DocumentDB Elastic Clusters, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon DocumentDB Elastic Clusters definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern nicht anders definiert, können nur Amazon DocumentDB Elastic Clusters diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden. Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dadurch werden Ihre Amazon DocumentDB Elastic Cluster-Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS Services, die mit IAM funktionieren. Suchen Sie in der Spalte Service-Linked Role nach den Services, die mit Ja markiert sind. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Dienstbezogene Rollenberechtigungen für elastische Cluster

Amazon DocumentDB Elastic Clusters verwendet die benannte serviceverknüpfte RolleAWS `ServiceRoleForDocDB-Elastic`, um es Amazon DocumentDB Elastic Clusters zu ermöglichen, AWS Dienste im Namen Ihrer Cluster aufzurufen.

Dieser dienstgebundenen Rolle ist eine Berechtigungsrichtlinie namens `AmazonDocDB-ElasticServiceRolePolicy` zugeordnet, die ihr Berechtigungen für den Betrieb in Ihrem Konto erteilt. Die Rollenberechtigungsrichtlinie ermöglicht es Amazon DocumentDB Elastic Clusters, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "AWS/DocDB-Elastic"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Wenn Sie auf die folgende Fehlermeldung stoßen: „Die Ressource konnte nicht erstellt werden. Überprüfen Sie, ob Sie die Berechtigung haben, eine serviceverknüpfte Rolle zu

erstellen. Andernfalls warten Sie und versuchen Sie es später erneut.“ , stellen Sie sicher, dass Sie die folgenden Berechtigungen aktiviert haben:

```
{
"Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/docdb-elastic.amazonaws.com/
AWSServiceRoleForDocDB-Elastic",
    "Condition": {
"StringLike": {
"iam:AWSServiceName":"docdb-elastic.amazonaws.com"
        }
    }
}
```

Weitere Informationen finden Sie unter Dienstbezogene Rollenberechtigungen im AWS Identity and Access Management-Benutzerhandbuch.

Eine serviceverknüpfte Rolle für elastische Amazon DocumentDB-Cluster erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine DB-Instance erstellen, erstellt Amazon DocumentDB Elastic Clusters die serviceverknüpfte Rolle für Sie.

Bearbeiten einer serviceverknüpften Rolle für elastische Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster ermöglichen es Ihnen nicht, die `AWS ServiceRoleForDocDB-Elastic` serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer dienstbezogenen Rolle im AWS Identity and Access Management-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon DocumentDB Elastic Clusters

Wenn Sie ein Feature oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch alle Ihre -Cluster löschen, bevor Sie die serviceverknüpfte Rolle löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

Um zu überprüfen, ob die serviceverknüpfte Rolle eine aktive Sitzung in der IAM-Konsole hat:

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole.
2. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus. Wählen Sie dann den Namen (nicht das Kontrollkästchen) der Rolle `AWS ServiceRoleForDocDB-Elastic` aus.
3. Wählen Sie auf der Seite Summary (Zusammenfassung) für die ausgewählte Rolle die Registerkarte Access Advisor (Advisor aufrufen) aus.

> **Note**
>
> Wenn Sie sich nicht sicher sind, ob Amazon DocumentDB Elastic Clusters die `AWS ServiceRoleForDocDB-Elastic` Rolle verwendet, können Sie versuchen, die Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt das Löschen fehl und Sie können sehen AWS-Regionen , wo die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.
> Wenn Sie die `AWS ServiceRoleForDocDB-Elastic` Rolle entfernen möchten, müssen Sie zuerst alle Ihre Cluster löschen.

Löschen aller Ihrer Cluster

Um einen Cluster in der Amazon DocumentDB DocumentDB-Konsole zu löschen:

1. Melden Sie sich bei der an AWS Management Consoleund öffnen Sie die Amazon DocumentDB DocumentDB-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, den Sie löschen möchten.
4. Klicken Sie bei Actions auf Delete.
5. Wenn Sie aufgefordert werden, den endgültigen Snapshot zu erstellen? , wählen Sie Ja oder Nein.

6. Wenn Sie im vorherigen Schritt Yes (Ja) gewählt haben, geben Sie unter Final snapshot name (Endgültiger Snapshot-Name) den Namen Ihres endgültigen DB-Snapshots ein.
7. Wählen Sie Delete (Löschen).

> **Note**
>
> Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um die serviceverknüpfte Rolle `AWS ServiceRoleForDocDB-Elastic` zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im AWS Identity and Access Management-Benutzerhandbuch.

Überwachung von Amazon DocumentDB

Die Überwachung Ihrer AWS Dienste ist ein wichtiger Bestandteil, damit Ihre Systeme gesund bleiben und optimal funktionieren. Es ist ratsam, Überwachungsdaten aus allen Teilen Ihrer AWS Lösung zu sammeln, damit Sie Fehler oder Beeinträchtigungen leichter debuggen und beheben können, falls sie auftreten sollten. Bevor Sie mit der Überwachung Ihrer AWS Lösungen beginnen, empfehlen wir Ihnen, die folgenden Fragen zu prüfen und Antworten darauf zu formulieren:

- Was sind Ihre Überwachungsziele?
- Welche Ressourcen werden Sie überwachen?
- Wie oft werden Sie diese Ressourcen überwachen?
- Welche Überwachungstools werden verwendet?
- Wer ist für die Überwachung verantwortlich?
- Wer ist zu benachrichtigen und auf welche Weise, wenn etwas schief geht?

Um Ihre aktuellen Leistungsmuster zu verstehen, Leistungsanomalien zu identifizieren und Methoden zur Behebung von Problemen zu formulieren, sollten Sie grundlegende Leistungskennzahlen für verschiedene Zeiten und unter unterschiedlichen Lastbedingungen festlegen. Bei der Überwachung Ihrer AWS Lösung empfehlen wir Ihnen, Ihre historischen Monitoring-Daten zu speichern, damit Sie future darauf zurückgreifen und Ihre Baselines erstellen können.

Die zulässigen Werte für Leistungsmetriken sind im Allgemeinen davon abhängig, wie die Ausgangsbasis aussieht und wofür die Anwendung gedacht ist. Prüfen Sie, ob dauerhafte oder tendenzielle Abweichungen von Ihrer Ausgangsbasis vorliegen. Im Folgenden werden Hinweise zu bestimmten Arten von Kennzahlen gegeben:

- Hohe CPU- oder RAM-Auslastung — Hohe Werte für die CPU- oder RAM-Auslastung können angemessen sein, vorausgesetzt, sie entsprechen Ihren Zielen für Ihre Anwendung (wie Durchsatz oder Parallelität) und werden erwartet.
- Speichervolumenverbrauch — Untersuchen Sie den Speicherverbrauch (`VolumeBytesUsed`), wenn der genutzte Speicherplatz konstant 85 Prozent des gesamten Speichervolumens ausmacht oder mehr beträgt. Legen Sie fest, ob Sie Daten aus dem Speicher-Volume löschen oder Daten in ein anderes System archivieren können, um Platz zu schaffen. Weitere Informationen erhalten Sie unter Amazon DocumentDB DocumentDB-Speicher und Amazon DocumentDB Kontingente und Beschränkungen.

- Netzwerkverkehr — Wenden Sie sich bezüglich des Netzwerkverkehrs an Ihren Systemadministrator, um zu erfahren, wie hoch der zu erwartende Durchsatz für Ihr Domänennetzwerk und Ihre Internetverbindung ist. Überprüfen Sie den Netzwerkdatenverkehr, wenn der Durchsatz dauerhaft unter dem erwarteten Wert liegt.
- Datenbankverbindungen — Erwägen Sie, Datenbankverbindungen einzuschränken, wenn Sie eine hohe Anzahl von Benutzerverbindungen in Verbindung mit einer verringerten Instanzleistung und Reaktionszeit feststellen. Die optimale Anzahl der Benutzerverbindungen für Ihre Instance ist von der Instance-Klasse und der Komplexität der Operationen abhängig, die ausgeführt werden.
- IOPS-Metriken — Die erwarteten Werte für IOPS-Metriken hängen von der Festplattenspezifikation und der Serverkonfiguration ab. Verwenden Sie daher Ihren Basiswert, um herauszufinden, was typisch ist. Prüfen Sie, ob dauerhafte Abweichungen von den Werten Ihrer Ausgangsbasis vorliegen. Für eine optimale IOPS-Leistung stellen Sie sicher, dass Ihr typisches Working Set in den Speicher passt, um Lese- und Schreibvorgänge zu minimieren.

Amazon DocumentDB (mit MongoDB-Kompatibilität) bietet eine Vielzahl von CloudWatch Amazon-Metriken, die Sie überwachen können, um den Zustand und die Leistung Ihrer Amazon DocumentDB-Cluster und -Instances zu ermitteln. Sie können Amazon DocumentDB-Metriken mit verschiedenen Tools anzeigen, darunter der Amazon DocumentDB DocumentDB-Konsole AWS CLI, CloudWatch API und Performance Insights.

Themen

- Den Status eines Amazon DocumentDB-Clusters überwachen
- Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen
- Amazon DocumentDB DocumentDB-Empfehlungen anzeigen
- Amazon DocumentDB DocumentDB-Veranstaltungsabonnements verwenden
- Überwachen von Amazon DocumentDB mit CloudWatch
- Protokollieren Amazon DocumentDB DocumentDB-API-Aufrufen mit AWS CloudTrail
- Profilierung von Amazon DocumentDB DocumentDB-Vorgängen
- Überwachung mit Performance Insights

Den Status eines Amazon DocumentDB-Clusters überwachen

Der Status eines Clusters zeigt den Zustand des Clusters an. Sie können den Status eines Clusters mithilfe der Amazon DocumentDB DocumentDB-Konsole oder des AWS CLI `describe-db-clusters` Befehls anzeigen.

Themen

- Werte für den Cluster-Status
- Den Status eines Clusters überwachen

Werte für den Cluster-Status

In der folgenden Tabelle sind die gültigen Werte für den Cluster-Status aufgeführt.

Cluster-Status	Beschreibung
`active`	Der Cluster ist aktiv. Dieser Status gilt nur für elastische Cluster.
`available`	Der Cluster ist stabil und verfügbar. Dieser Status gilt nur für instanzbasierte Cluster.
`backing-up`	Der Cluster wird derzeit gesichert.
`creating`	Der Cluster wird gerade erstellt. Ist während der Erstellung unzugänglich.
`deleting`	Der Cluster wird gerade gelöscht. Ist während des Löschvorgangs nicht zugreifbar.
`failing-over`	Ein Failover von der primären Instance zu einem Amazon

Cluster-Status	Beschreibung
	DocumentDB DocumentDB-Replikat wird durchgeführt.
`inaccessible-encryption-credentials`	Auf den AWS KMS Schlüssel , der zum Verschlüsseln oder Entschlüsseln des Clusters verwendet wurde, kann nicht zugegriffen werden.
`maintenance`	Ein Wartungsupdate wird auf den Cluster angewende t. Dieser Status wird für Wartungsarbeiten auf Clusterebene verwendet, die Amazon DocumentDB weit im Voraus plant.
`migrating`	Es wird gerade ein Cluster-Snapshot in einen Cluster wiederhergestellt.
`migration-failed`	Eine Migration ist fehlgesch lagen.
`modifying`	Der Cluster wird aufgrund einer Kundenanfrage, den Cluster zu ändern, geändert.
`renaming`	Der Cluster wird aufgrund einer Kundenanfrage auf Instance-Umbenennung umbenannt.
`resetting-master-credentials`	Die Master-Anmeldeinfo rmationen für den Cluster werden auf eine Kundenanf rage auf Zurücksetzung hin zurückgesetzt.

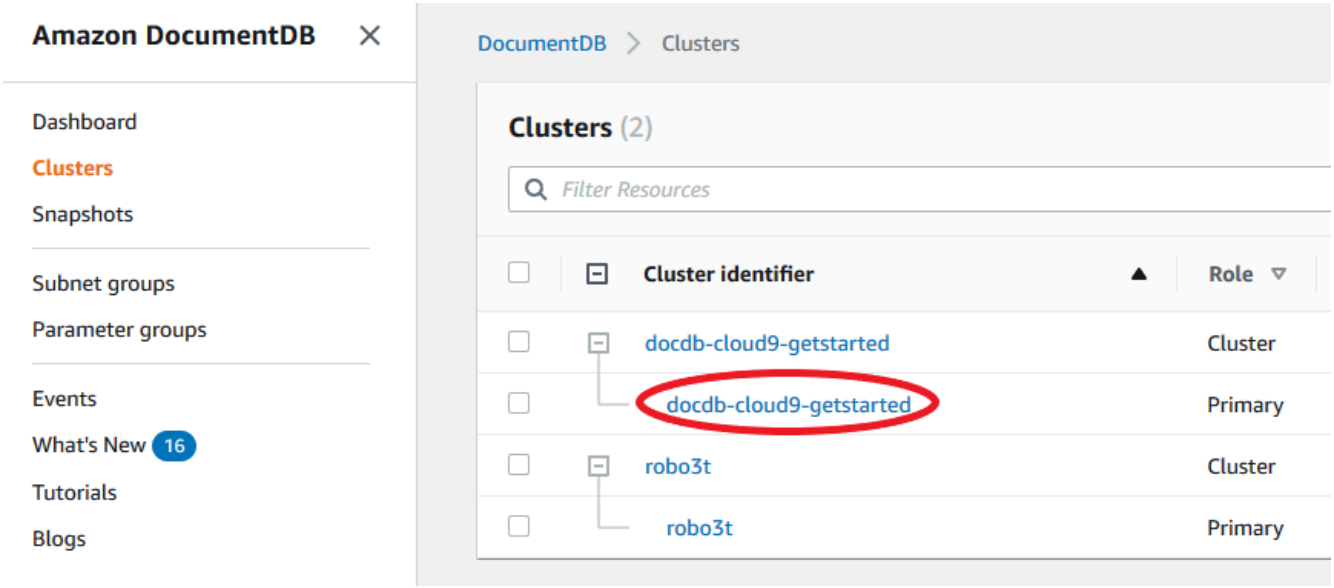
Cluster-Status	Beschreibung
`upgrading`	Die Cluster-Engine wird auf eine neue Version aktualisiert.

Den Status eines Clusters überwachen

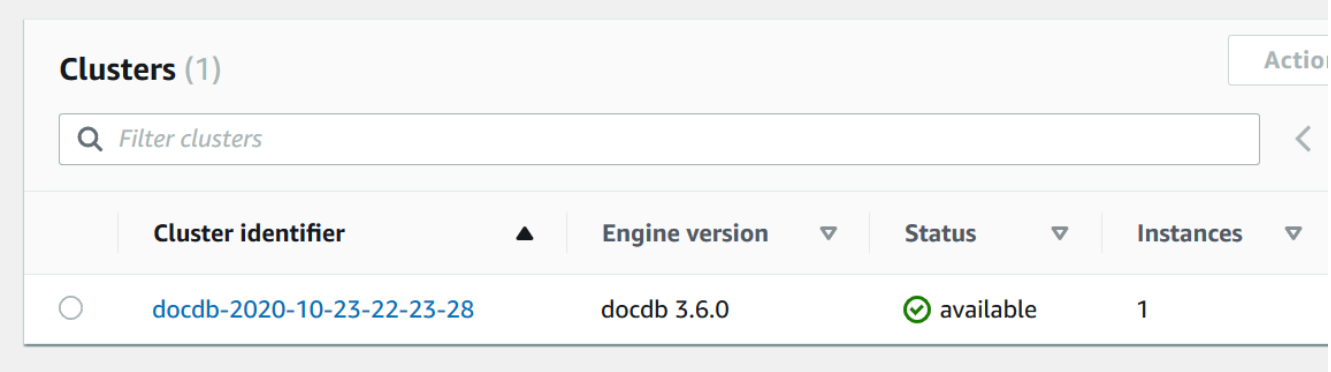
Using the AWS Management Console

Verwenden Sie das folgende Verfahren, AWS Management Console um den Status eines Clusters zu ermitteln.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Suchen Sie in der Spalte Cluster-ID den Namen der Instance, an der Sie interessiert sind. Um den Status der Instance zu ermitteln, lesen Sie dann über diese Zeile bis zur Spalte Status, wie unten gezeigt.



Using the AWS CLI

Wenn Sie AWS CLI den Status eines Clusters ermitteln möchten, verwenden Sie den `describe-db-clusters` Vorgang. Der folgende Code ermittelt den Status des Clusters `sample-cluster`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster  \
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Für Windows:

```
aws docdb describe-db-clusters ^
    --db-cluster-identifier sample-cluster  ^
    --query 'DBClusters[*].[DBClusterIdentifier,Status]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "sample-cluster",
        "available"
    ]
]
```

Den Status einer Amazon DocumentDB DocumentDB-Instance überwachen

Amazon DocumentDB bietet Informationen über den aktuellen Zustand jeder konfigurierten Instanz in der Datenbank.

Es gibt drei Statustypen, die Sie für eine Amazon DocumentDB DocumentDB-Instance anzeigen können:

- Instance-Status: Dieser Status wird in der Spalte Status der Cluster-Tabelle im angezeigt AWS Management Console und zeigt den aktuellen Lebenszykluszustand der Instance an. Die in der Spalte Status angezeigten Werte werden aus dem `Status` Feld der `DescribeDBCluster` API-Antwort abgeleitet.
- Integritätsstatus der Instanz: Dieser Status wird in der Spalte Instanzstatus der Clustertabelle im angezeigt AWS Management Console und gibt an, ob die Datenbank-Engine, die Komponente, die für die Verwaltung und das Abrufen von Daten verantwortlich ist, läuft. Die in der Spalte Instance-Zustand angezeigten Werte basieren auf der CloudWatch `EngineUptime` Amazon-Systemmetrik.
- Wartungsstatus: Dieser Status wird in der Spalte Wartung der Cluster-Tabelle im angezeigt AWS Management Console und gibt den Status aller Wartungsereignisse an, die auf eine Instance angewendet werden müssen. Der Wartungsstatus ist unabhängig vom Status der anderen Instanzen und wird von der `PendingMaintenanceAction` API abgeleitet. Weitere Informationen zum Wartungsstatus finden Sie unter Wartung von Amazon DocumentDB.

Themen

- Instance-Statuswerte
- Überwachen Sie den Instance-Status mit dem AWS Management Console oder AWS CLI
- Werte für den Integritätsstatus der Instanz
- Überwachen des Zustands der Instanz mit dem AWS Management Console

Instance-Statuswerte

Die folgende Tabelle führt die möglichen Statuswerte für Instances auf und gibt an, wie welcher Status berechnet wird. Die Tabelle gibt an, ob Sie für Instance und Speicher, nur für Speicher oder weder für Instance noch Speicher zahlen müssen. Für alle Instance-Status wird immer die Sicherungsnutzung berechnet.

Instance-Status	Berechnet	Beschreibung
`available`	Berechnet	Die Instance ist stabil und verfügbar.
`backing-up`	Berechnet	Die Instance wird derzeit gesichert.
`configuring-log-exports`	Berechnet	Das Veröffentlichen von Protokolldateien in Amazon CloudWatch Logs wird für diese Instance aktiviert oder deaktiviert.
`creating`	Nicht berechnet	Die Instance wird gerade erstellt. Während die Instance erstellt wird, kann nicht auf sie zugegriffen werden.
`deleting`	Nicht berechnet	Die Instance wird gerade gelöscht.
`failed`	Nicht berechnet	Die Instance ist ausgefallen und Amazon DocumentDB konnte sie nicht wiederher stellen. Um die Daten wiederherzustellen , führen Sie eine point-in-time Wiederher stellung zum letzten wiederherstellbaren Zeitpunkt der Instance durch.
`inaccessible-encryption-credentials`	Nicht berechnet	Auf den AWS KMS Schlüssel, der zum Verschlüsseln oder Entschlüsseln der Instanz verwendet wird, konnte nicht zugegriffen werden.
`incompatible-network`	Nicht berechnet	Amazon DocumentDB versucht, eine Wiederherstellungsaktion für eine Instance durchzuführen, kann dies jedoch nicht tun, da sich die VPC in einem Zustand befindet, der verhindert, dass die Aktion abgeschlossen werden kann. Dieser Status kann auftreten, wenn beispielsweise alle verfügbaren IP-Adressen in einem Subnetz verwendet wurden und Amazon DocumentD

Instance-Status	Berechnet	Beschreibung
		B keine IP-Adresse für die Instance abrufen konnte.
`maintenance`	Berechnet	Amazon DocumentDB wendet ein Wartungsupdate auf die Instance an. Dieser Status wird für Wartungsarbeiten auf Instanzebene verwendet, die Amazon DocumentDB weit im Voraus plant. Wir evaluieren derzeit Möglichkeiten, Kunden über diesen Status auch weitere Wartungsa ktionen kenntlich zu machen.
`modifying`	Berechnet	Die Instance wird aufgrund einer Anforderu ng zur Änderung der Instance geändert.
`rebooting`	Berechnet	Die Instance wird aufgrund einer Anfrage oder eines Amazon DocumentDB DocumentDB-Prozesses, der einen Neustart der Instance erfordert, neu gestartet.
`renaming`	Berechnet	Die Instance wird aufgrund einer Anforderu ng zur Umbenennung umbenannt.
`resetting -master-c redentials`	Berechnet	Die Master-Anmeldeinformationen für die Instance werden aufgrund einer Anforderung zum Zurücksetzen zurückgesetzt.
`restore-error`	Berechnet	Bei dem Versuch, einen oder aus einem Snapshot wiederherzustellen, ist bei der Instance ein point-in-time Fehler aufgetret en.
`starting`	Berechnet für Speicher	Die Instance wird gestartet.
`stopped`	Berechnet für Speicher	Die Instance wird angehalten.
`stopping`	Berechnet für Speicher	Die Instance wird gestoppt.

Instance-Status	Berechnet	Beschreibung
`storage-full`	Berechnet	Die Instance hat seine Speicherkapazitäts zuteilung erreicht. Dieser Status ist kritisch und sollte sofort behoben werden. Sie müssen den verfügbaren Speicherplatz hochskalieren, indem Sie die Instance entsprechend ändern. Stellen Sie CloudWatch Amazon-Alarme so ein, dass Sie gewarnt werden, wenn der Speicherplatz knapp wird, damit Sie nicht in diese Situation geraten.

Überwachen Sie den Instance-Status mit dem AWS Management Console oder AWS CLI

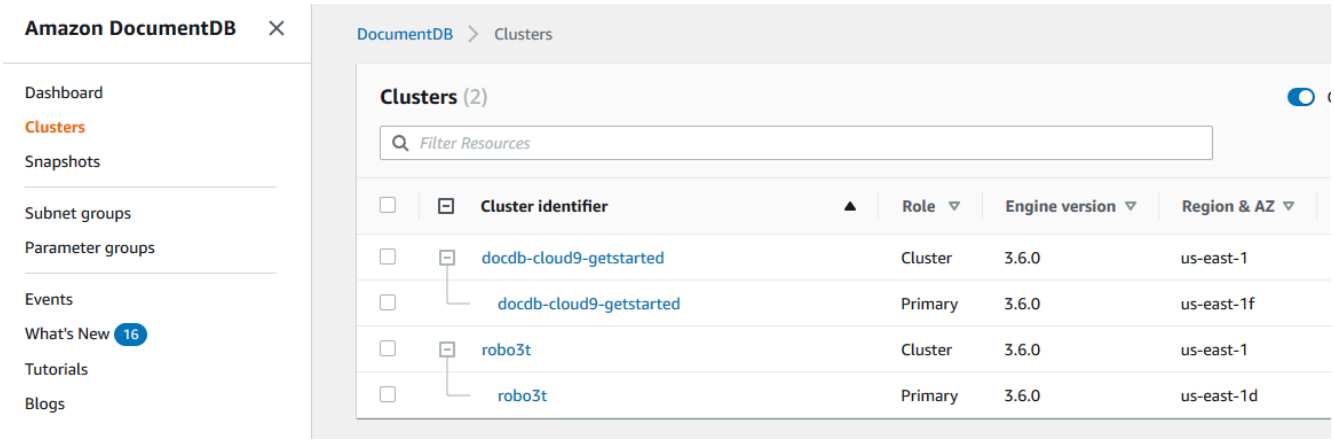
Verwenden Sie das AWS Management Console oder AWS CLI , um den Status Ihrer Instanz zu überwachen.

Using the AWS Management Console

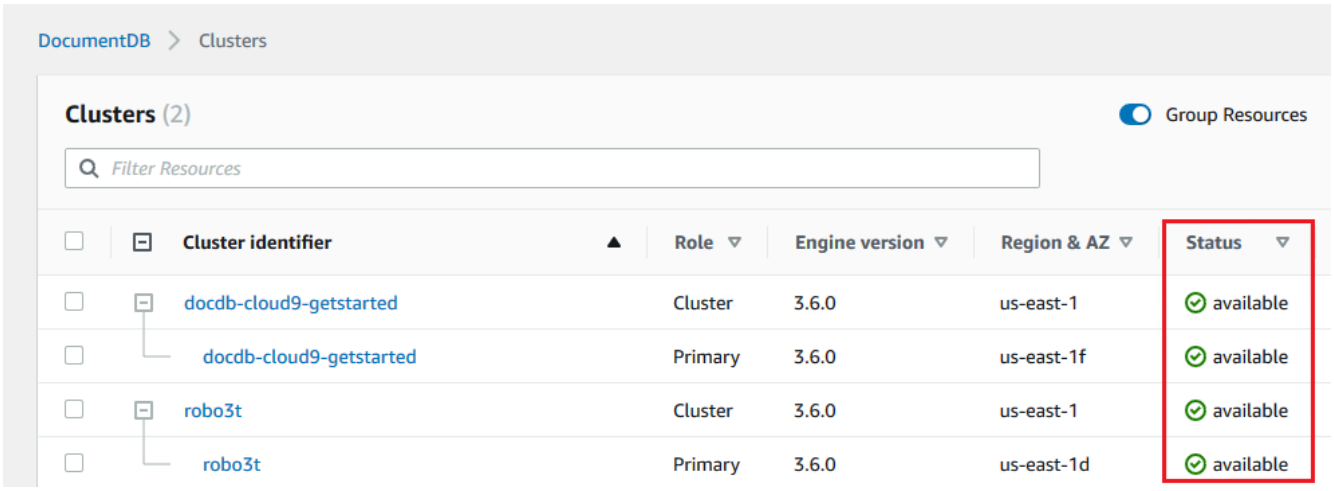
Verwenden Sie das folgende Verfahren, AWS Management Console um den Status eines Clusters zu ermitteln.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Note**
>
> Beachten Sie, dass im Cluster-Navigationsfeld in der Spalte Cluster-ID sowohl Cluster als auch Instances angezeigt werden. Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



3. Suchen Sie den Namen der Instance, an der Sie interessiert sind. Um den Status der Instance zu ermitteln, prüfen Sie die Zeile in die Spalte Status.



Using the AWS CLI

Wenn Sie AWS CLI den Status eines Clusters ermitteln möchten, verwenden Sie den `describe-db-instances` Vorgang. Der folgende Code ermittelt den Status der Instance `sample-cluster-instance-01`.

Für Linux, macOS oder Unix:

```
aws docdb describe-db-instances \
          --db-instance-identifier sample-cluster-instance-01  \
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Für Windows:

```
aws docdb describe-db-instances ^
          --db-instance-identifier sample-cluster-instance-01  ^
          --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
          [
              "sample-cluster-instance-01",
              "available"
          ]
      ]
```

Werte für den Integritätsstatus der Instanz

In der folgenden Tabelle sind die möglichen Integritätsstatuswerte für Instances aufgeführt. Die Spalte Instanzstatus in der Cluster-Tabelle in der zeigt AWS Management Console, ob die Datenbank-Engine, die Komponente, die für das Speichern, Verwalten und Abrufen von Daten verantwortlich ist, normal funktioniert. In dieser Spalte wird auch angegeben, ob die `EngineUptime` Systemmetrik CloudWatch, verfügbar in, den Integritätsstatus jeder Instanz anzeigt.

Instance-Zustand	Beschreibung
fehlerfrei	Die Datenbank-Engine läuft in der Amazon DocumentDB DocumentDB-Instance.
ungesund	Die Datenbank-Engine läuft nicht oder wurde vor weniger als einer Minute neu gestartet.

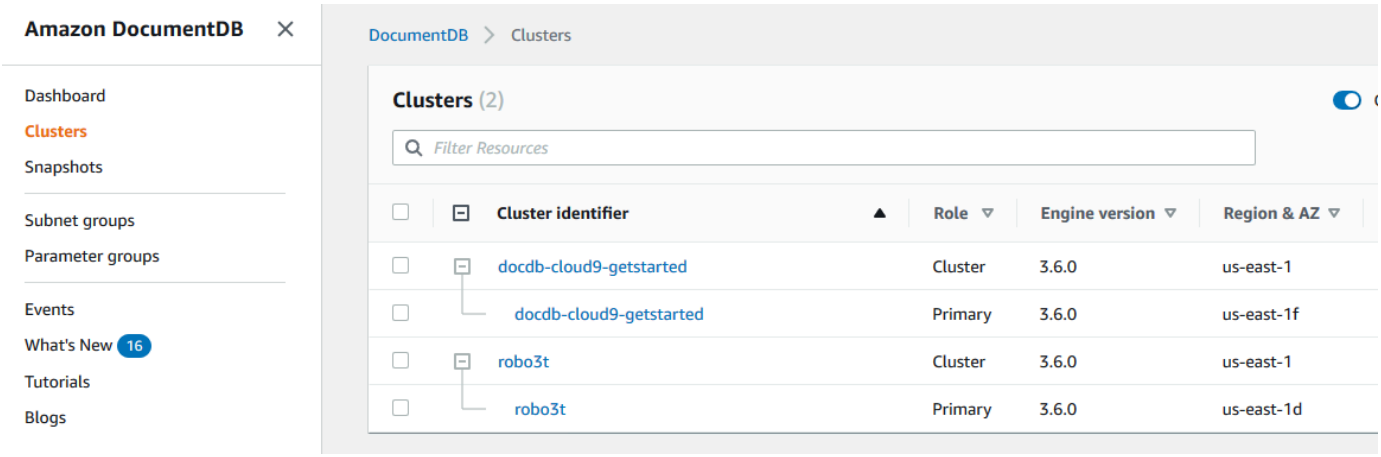
Überwachen des Zustands der Instanz mit dem AWS Management Console

Verwenden Sie den AWS Management Console , um den Integritätsstatus Ihrer Instance zu überwachen.

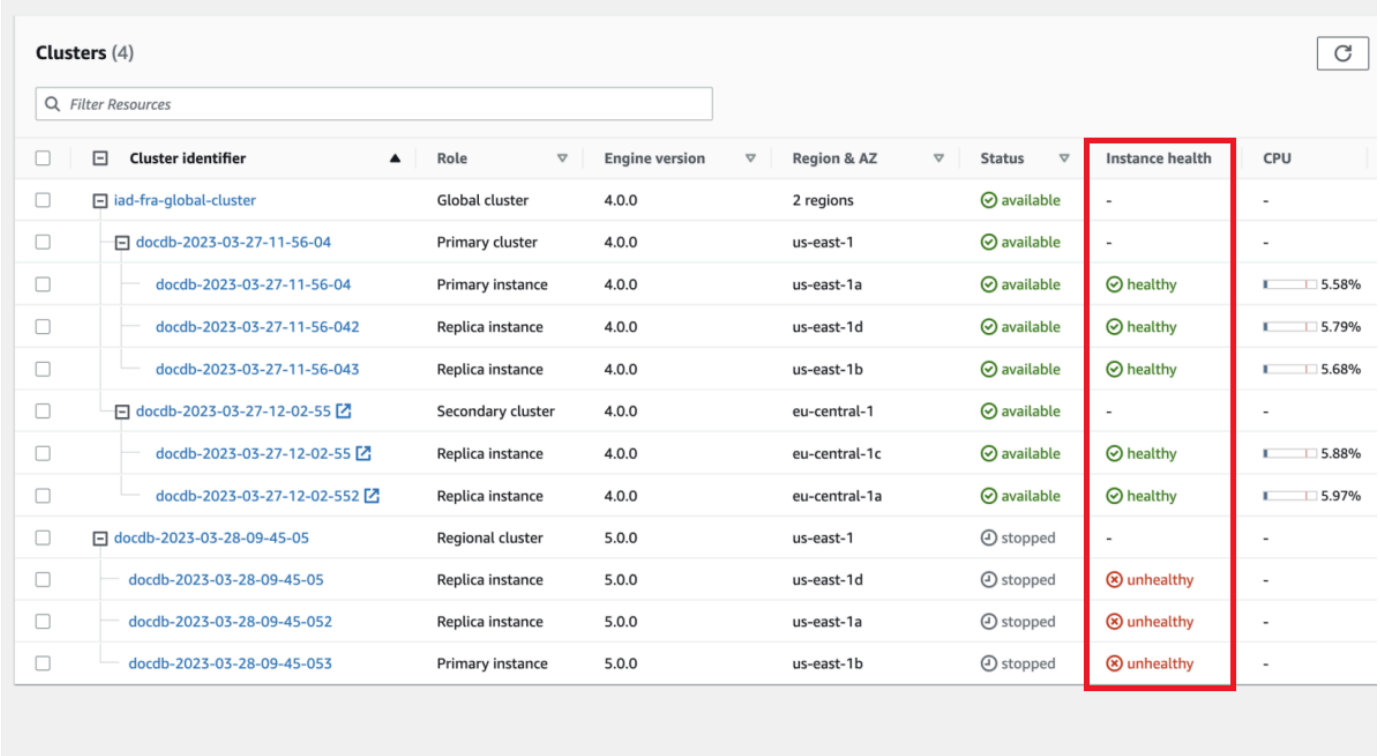
Verwenden Sie bei der Verwendung die folgenden Schritte AWS Management Console, um den Integritätsstatus der Instanz zu verstehen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

> **Note**
>
> Im Cluster-Navigationsfeld werden in der Spalte Cluster-ID sowohl Cluster als auch Instances angezeigt. Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



3. Suchen Sie den Namen der Instance, an der Sie interessiert sind. Um den Status der Instanz zu ermitteln, lesen Sie dann in dieser Zeile bis zur Spalte Instanzstatus nach, wie in der folgenden Abbildung dargestellt:



> **Note**
>
> Die Abfrage des Instance-Integritätsstatus erfolgt alle 60 Sekunden und basiert auf der CloudWatch `EngineUptime` Systemmetrik. Die Werte in der Spalte Instanzstatus werden automatisch aktualisiert.

Amazon DocumentDB DocumentDB-Empfehlungen anzeigen

Amazon DocumentDB bietet eine Liste automatisierter Empfehlungen für Datenbankressourcen wie Instances und Cluster. Diese Empfehlungen dienen als Leitfaden für bewährte Verfahren, indem Sie Ihre Cluster- und Instance-Konfigurationen analysieren.

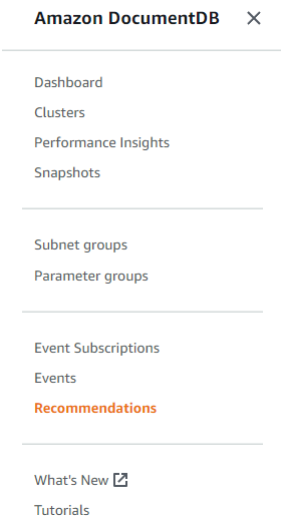
Ein Beispiel für diese Empfehlungen finden Sie im Folgenden:

Typ	Beschreibung	Empfehlung	Zusätzliche Informationen
Eine -Instance	Der Cluster enthält nur eine Instanz	Leistung und Verfügbarkeit: Wir empfehlen, eine weitere Instanz mit derselben Instanzklasse in einer anderen Availability Zone hinzuzufügen.	Amazon DocumentDB Hochverfügbarkeit und Replikation

Amazon DocumentDB generiert Empfehlungen für eine Ressource, wenn die Ressource erstellt oder geändert wird. Amazon DocumentDB scannt Ihre Ressourcen auch regelmäßig und generiert Empfehlungen.

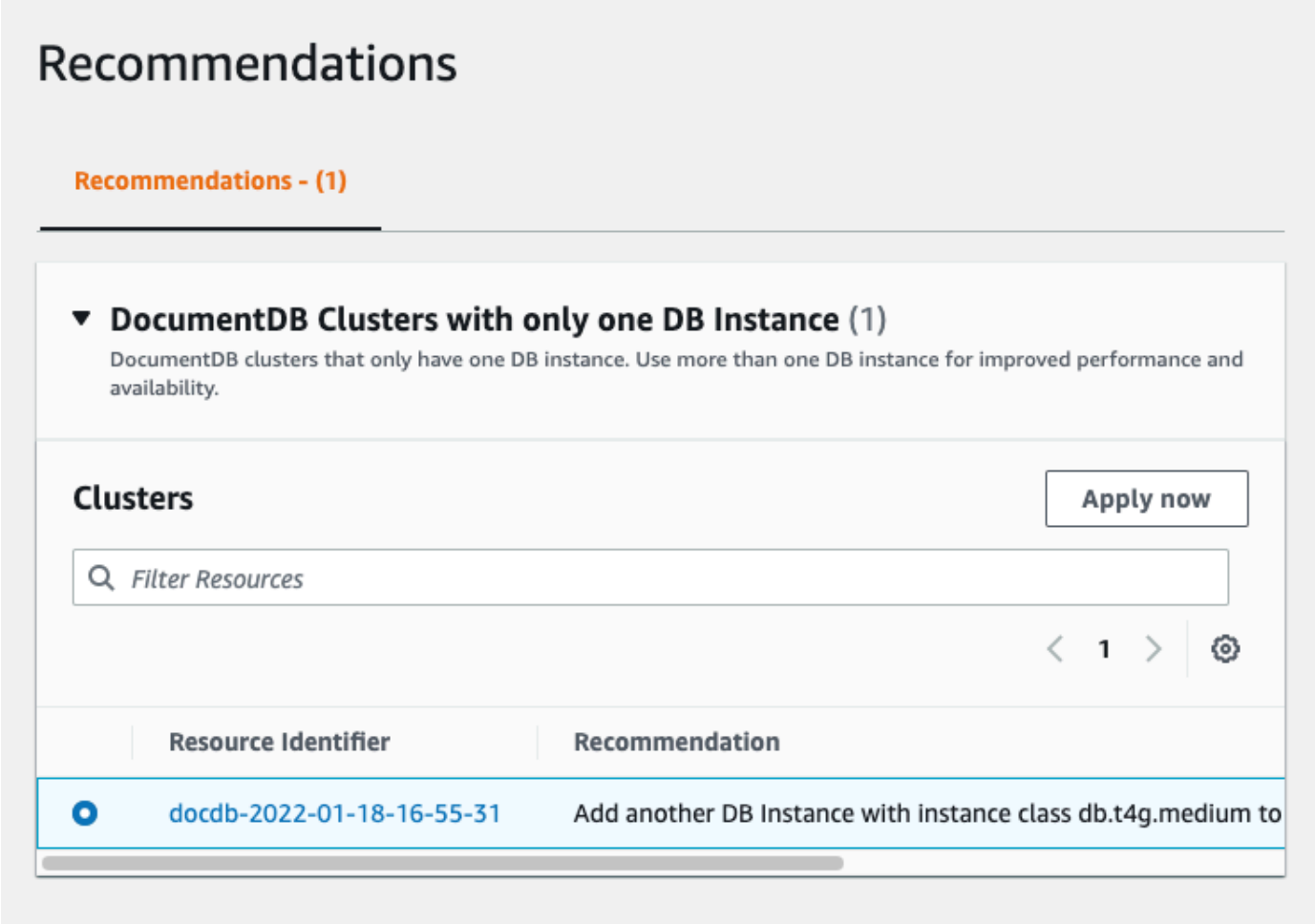
Um Empfehlungen von Amazon DocumentDB einzusehen und entsprechende Maßnahmen zu ergreifen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Empfehlungen aus:



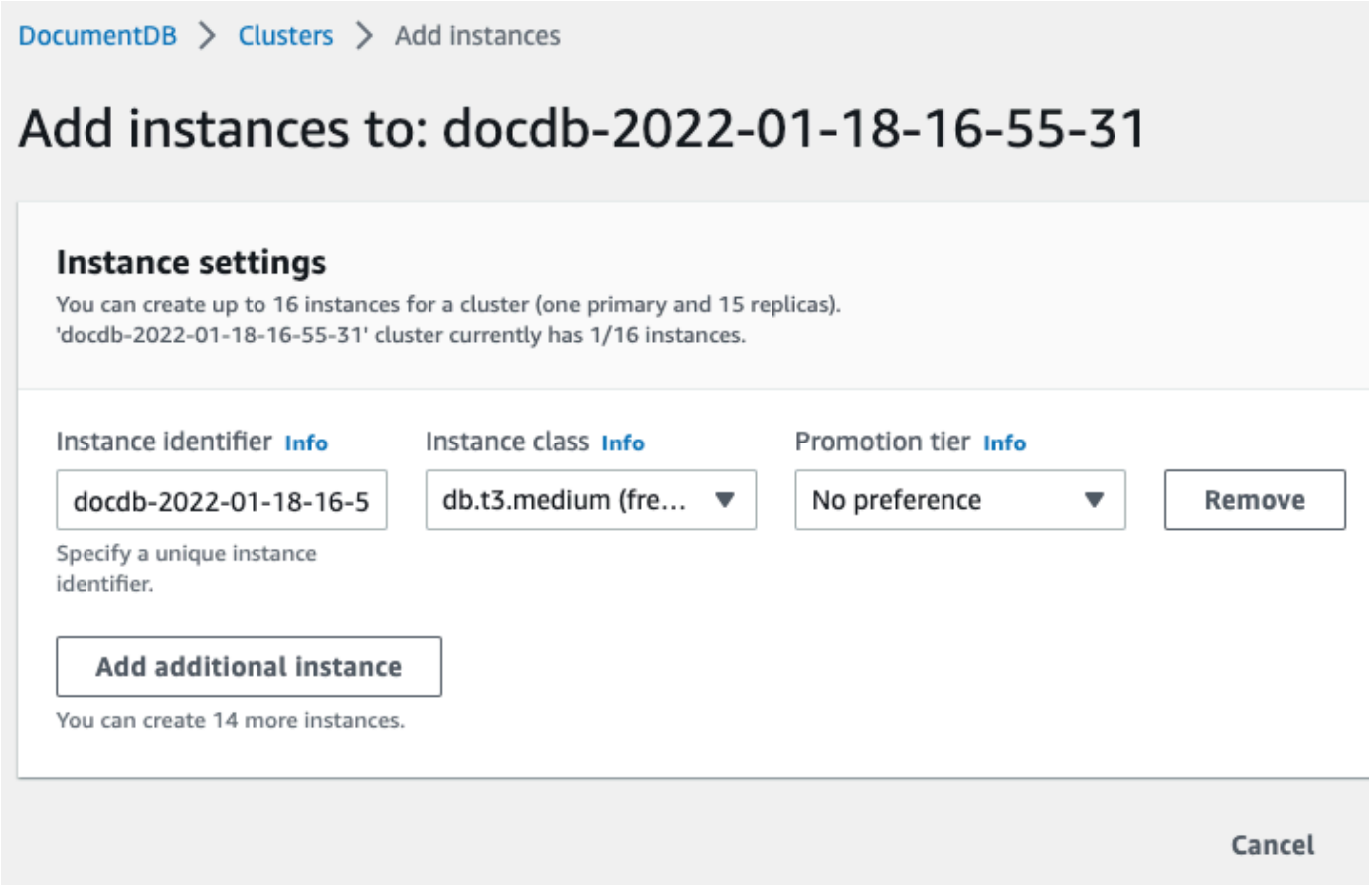
3. Erweitern Sie im Dialogfeld „Empfehlungen“ den gewünschten Bereich und wählen Sie die empfohlene Aufgabe aus.

Im folgenden Beispiel bezieht sich die empfohlene Aufgabe auf einen Amazon DocumentDB-Cluster mit nur einer Instance. Es wird empfohlen, eine weitere Instance hinzuzufügen, um die Leistung und Verfügbarkeit zu verbessern.



4. Klicken Sie auf Jetzt bewerben.

   In diesem Beispiel wird das Dialogfeld „Instanzen hinzufügen“ angezeigt:



5. Ändern Sie die Einstellungen Ihrer neuen Instanz und klicken Sie auf Erstellen.

Amazon DocumentDB DocumentDB-Veranstaltungsabonnements verwenden

Amazon DocumentDB verwendet Amazon Simple Notification Service (Amazon SNS), um Benachrichtigungen bereitzustellen, wenn ein Amazon DocumentDB DocumentDB-Ereignis eintritt. Diese Benachrichtigungen können in jeder Form erfolgen, die von Amazon SNS für eine unterstützt wird AWS-Region, z. B. eine E-Mail, eine Textnachricht oder ein Anruf an einen HTTP-Endpunkt.

Amazon DocumentDB gruppiert diese Ereignisse in Kategorien, die Sie abonnieren können, sodass Sie benachrichtigt werden können, wenn ein Ereignis in dieser Kategorie eintritt. Sie können eine Ereigniskategorie für eine Instance, einen Cluster, einen Snapshot, einen Cluster-Snapshot oder eine Parametergruppe abonnieren. Wenn Sie beispielsweise die Backup-Kategorie für eine bestimmte Instanz abonnieren, werden Sie jedes Mal benachrichtigt, wenn ein Backup-bezogenes Ereignis eintritt, das sich auf die Instanz auswirkt. Sie erhalten auch eine Benachrichtigung, wenn sich ein Abonnement für ein Ereignis ändert.

Ereignisse treten sowohl auf Cluster- als auch auf Instanzebene auf, sodass Sie Ereignisse empfangen können, wenn Sie einen Cluster oder eine Instance abonnieren.

Ereignisabonnements werden an die Adressen gesendet, die Sie bei der Erstellung des Abonnements angegeben haben. Möglicherweise möchten Sie mehrere verschiedene Abonnements erstellen, z. B. ein Abonnement, das alle Ereignisbenachrichtigungen empfängt, und ein anderes Abonnement, das nur kritische Ereignisse für Ihre Produktionsinstanzen enthält. Sie können die Benachrichtigungen ganz einfach deaktivieren, ohne ein Abonnement zu löschen. Setzen Sie dazu das Optionsfeld Aktiviert in der Amazon DocumentDB DocumentDB-Konsole auf Nein.

> **Important**
>
> Amazon DocumentDB garantiert nicht die Reihenfolge der Ereignisse, die in einem Event-Stream gesendet werden. Die Reihenfolge der Ereignisse kann sich ändern.

Amazon DocumentDB verwendet den Amazon-Ressourcennamen (ARN) eines Amazon SNS-Themas, um jedes Abonnement zu identifizieren. Die Amazon DocumentDB DocumentDB-Konsole erstellt den ARN für Sie, wenn Sie das Abonnement erstellen.

Die Abrechnung von Amazon DocumentDB DocumentDB-Veranstaltungsabonnements erfolgt über Amazon SNS. Bei Verwendung von Ereignisbenachrichtigungen fallen Amazon-SNS-Gebühren an. Weitere Informationen finden Sie unter Preise für den Amazon Simple Notification Service (SNS). Abgesehen von den Amazon SNS SNS-Gebühren stellt Amazon DocumentDB keine Event-Abonnements in Rechnung.

Themen

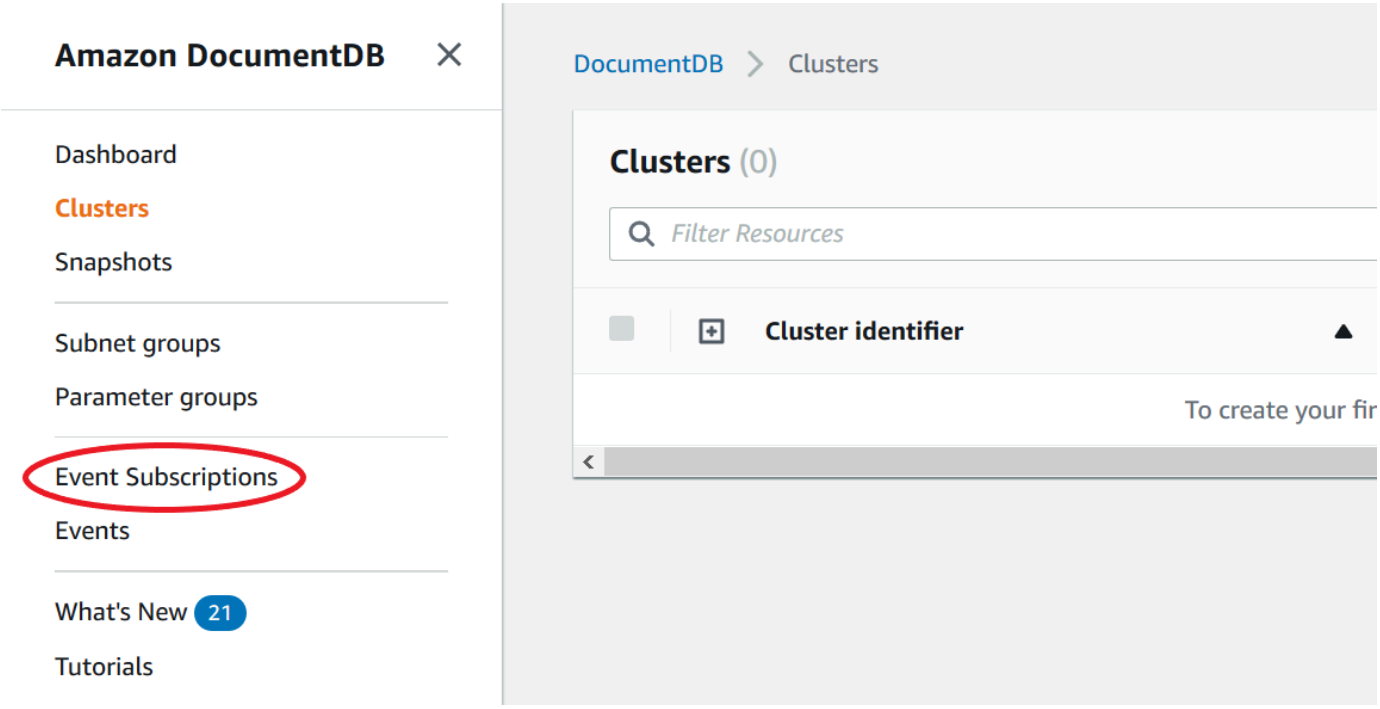
- Amazon DocumentDB DocumentDB-Veranstaltungen abonnieren
- Verwaltung von Abonnements für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen
- Amazon DocumentDB DocumentDB-Ereigniskategorien und Meldungen

Amazon DocumentDB DocumentDB-Veranstaltungen abonnieren

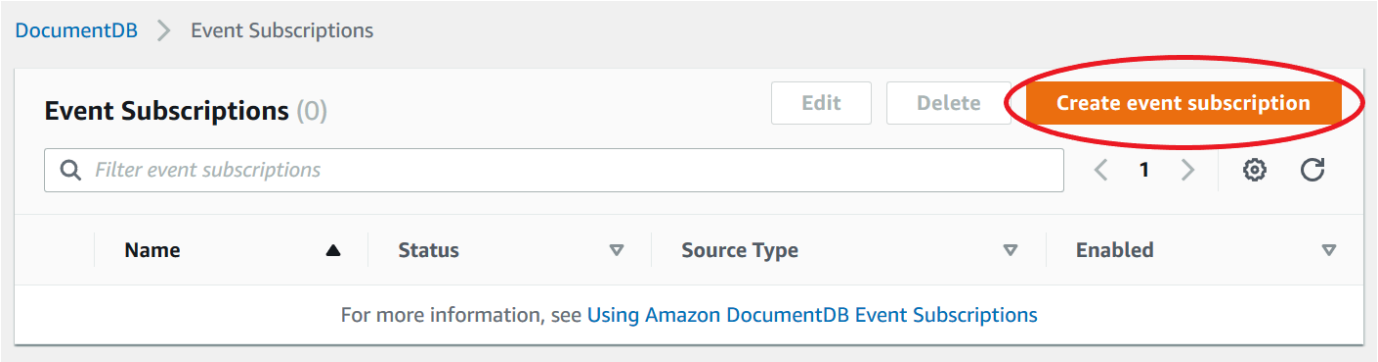
Sie können die Amazon DocumentDB DocumentDB-Konsole verwenden, um Veranstaltungsabonnements wie folgt zu abonnieren:

1. Melden Sie sich AWS Management Console unter an. https://console.aws.amazon.com/docdb

2. Wählen Sie im Navigationsbereich Ereignisabonnements aus.

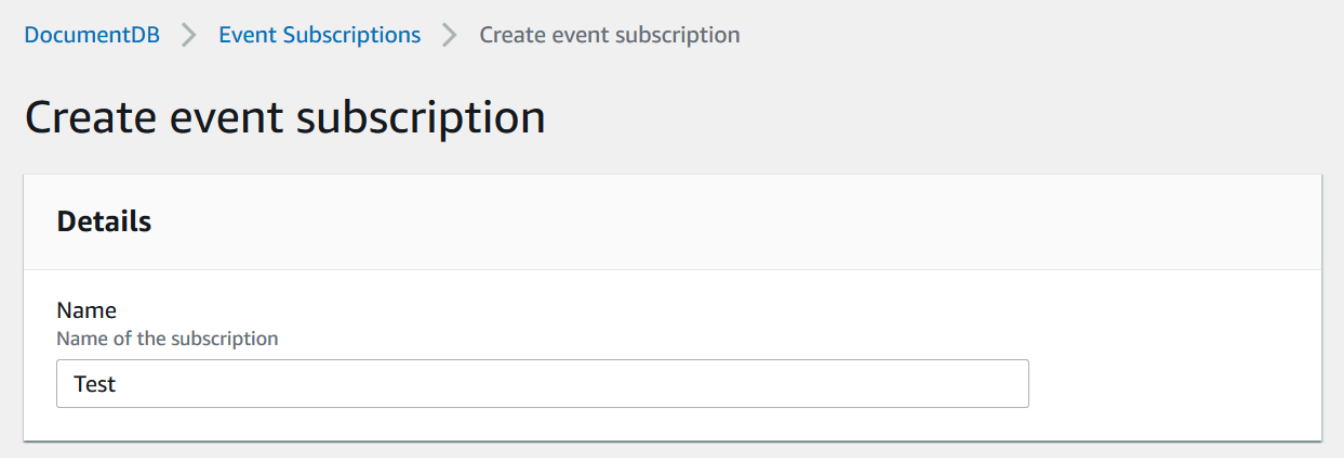


3. Wählen Sie im Bereich Ereignisabonnements Ereignisabonnement erstellen aus.

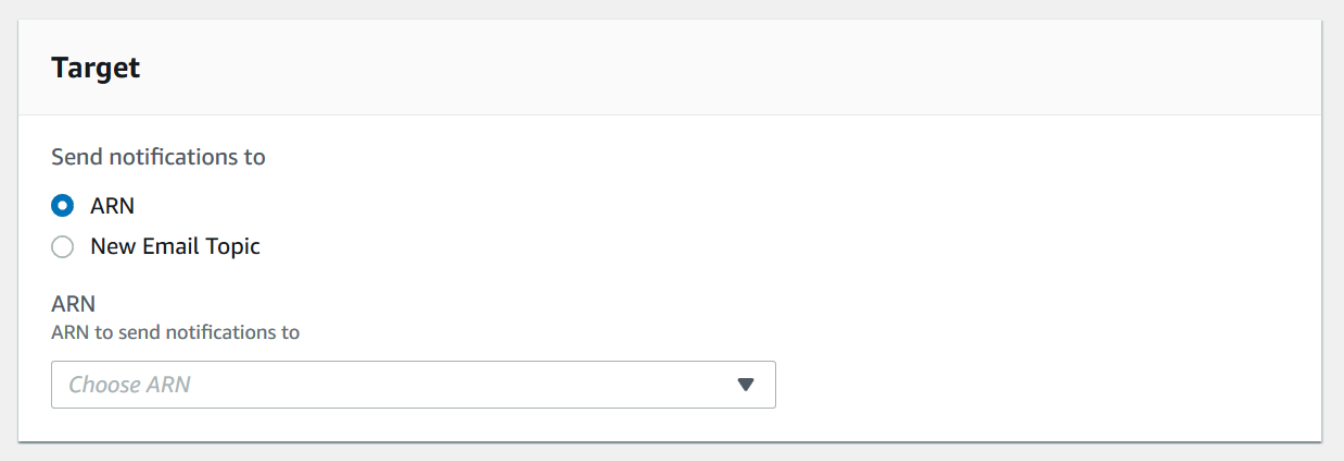


4. Gehen Sie im Dialogfeld Ereignisabonnement erstellen wie folgt vor:

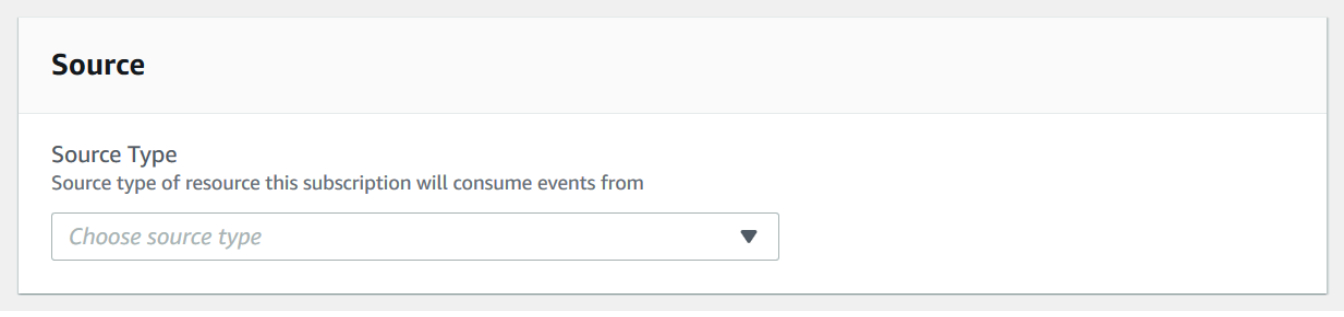
   - Geben Sie unter Name einen Namen für das Abonnement für Ereignisbenachrichtigungen ein.



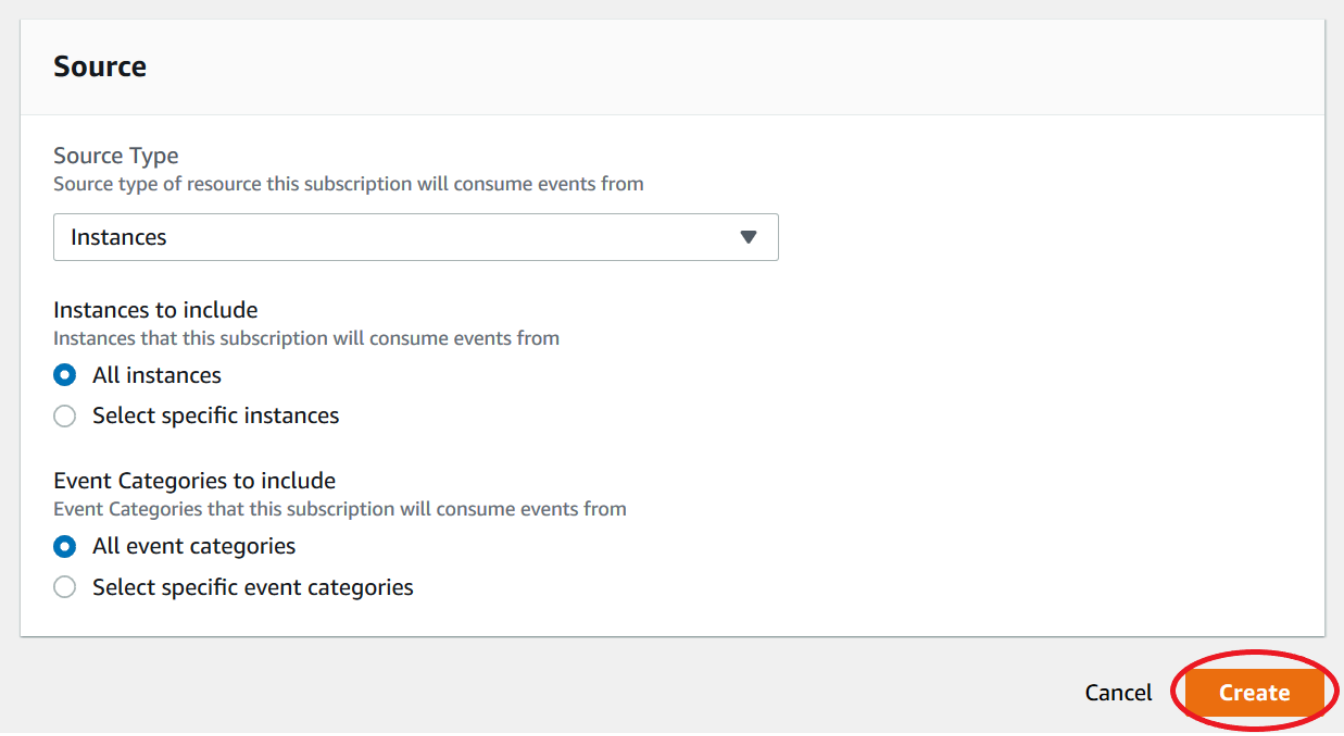
- Wählen Sie für Target aus, wohin Sie Benachrichtigungen senden möchten. Sie können einen vorhandenen ARN oder Neues E-Mail-Thema wählen, um den Namen eines Themas und eine Empfängerliste einzugeben.



- Wählen Sie für Quelle einen Quelltyp aus. Wählen Sie abhängig vom ausgewähltem Quelltyp die Ereigniskategorien und Quellen aus, für die Sie Ereignisbenachrichtigungen erhalten möchten.



- Wählen Sie Create (Erstellen) aus.

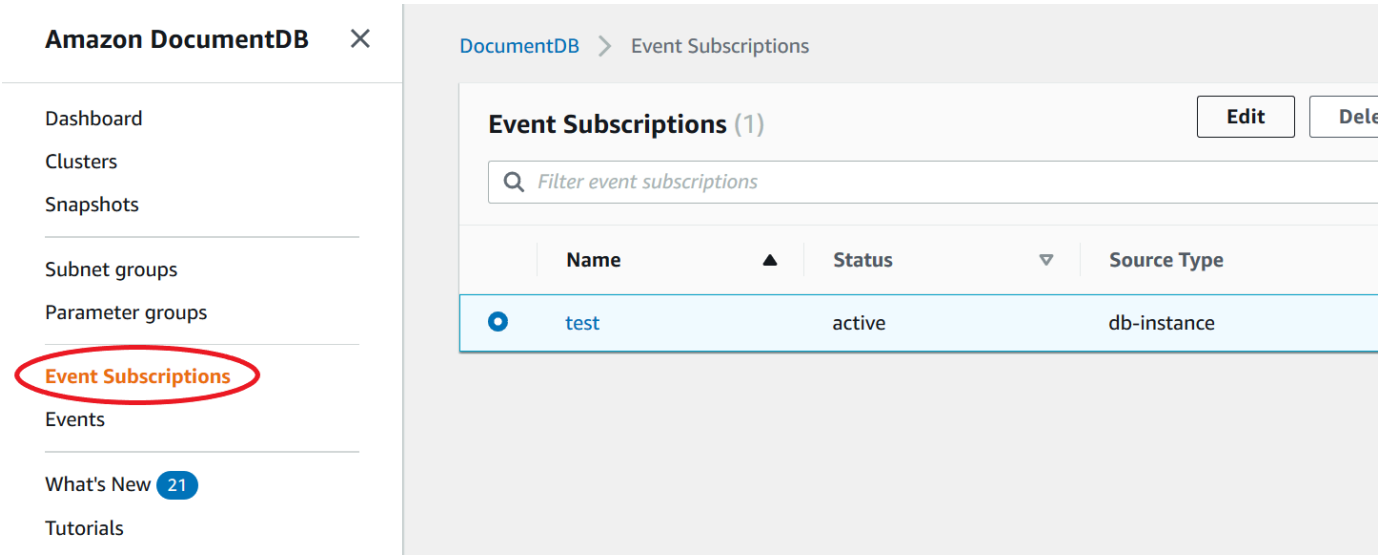


Verwaltung von Abonnements für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen

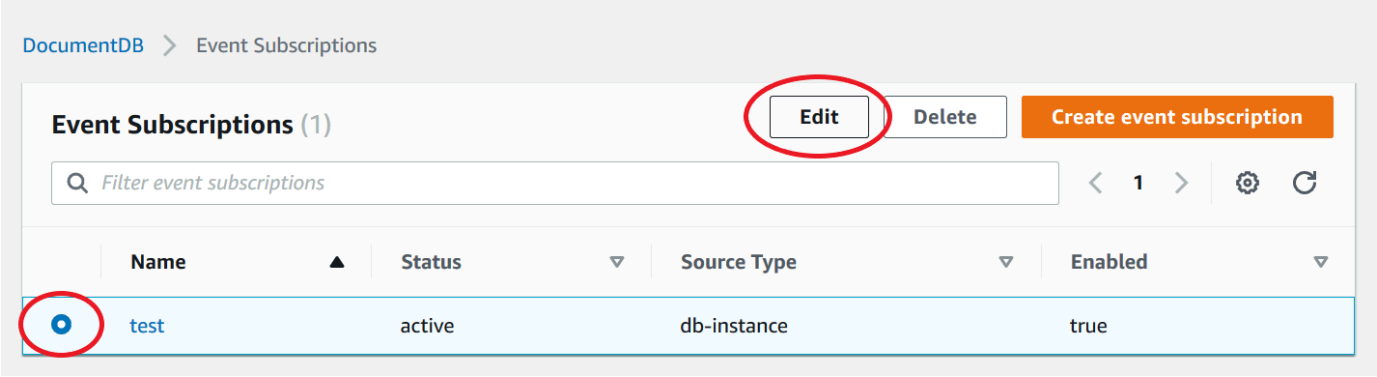
Wenn Sie im Navigationsbereich der Amazon DocumentDB DocumentDB-Konsole Event-Abonnements auswählen, können Sie Abonnementkategorien und eine Liste Ihrer aktuellen Abonnements einsehen. Sie können auch ein bestimmtes Abonnement ändern oder löschen.

So ändern Sie Ihre aktuellen Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen

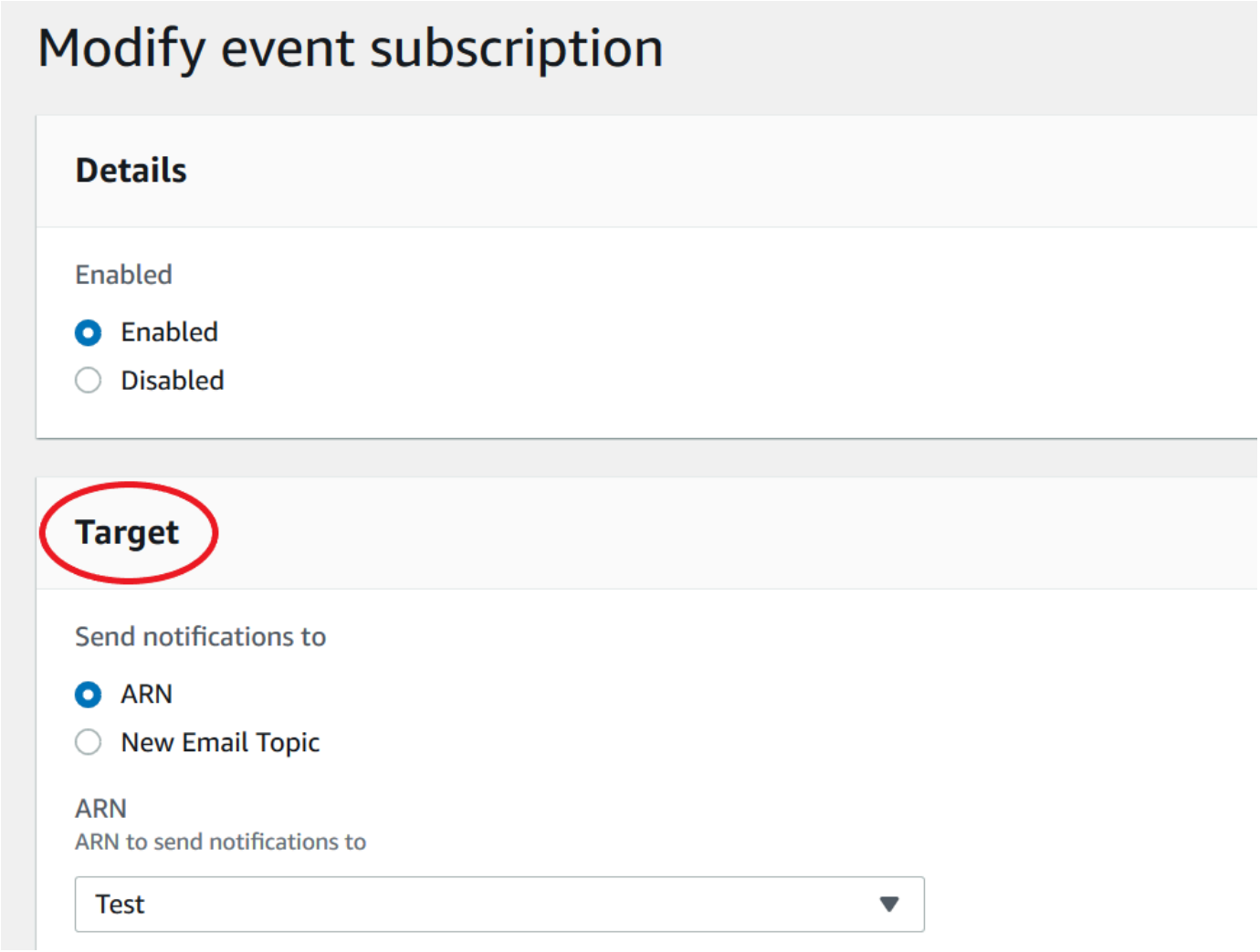
1. Melden Sie sich AWS Management Console unter an. https://console.aws.amazon.com/docdb
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus. Im Bereich Ereignisabonnements werden all Ihre Abonnements für Ereignisbenachrichtigungen angezeigt.



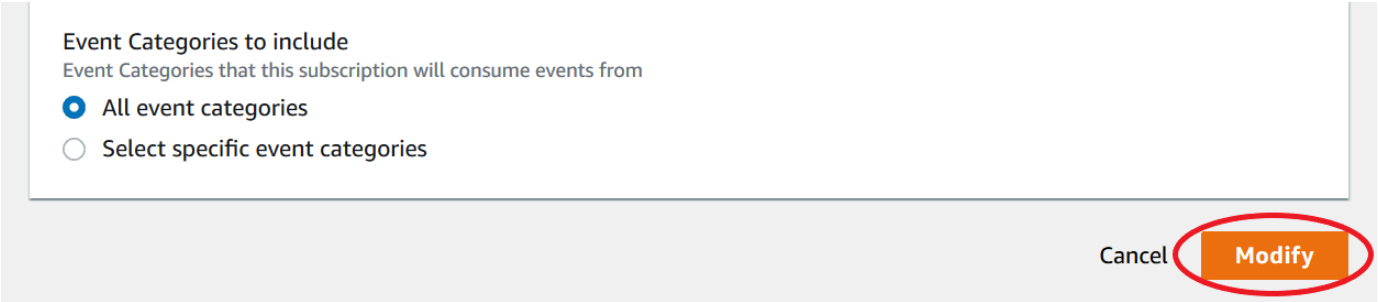
3. Wählen Sie im Bereich Ereignisabonnements das Abonnement, das Sie modifizieren möchten, und klicken Sie auf Bearbeiten.



4. Nehmen Sie Ihre Änderungen am Abonnement im Bereich Ziel oder Quelle vor. Sie können Quell-IDs hinzufügen oder entfernen, indem Sie diese im Bereich Quelle auswählen oder ihre Auswahl aufheben.



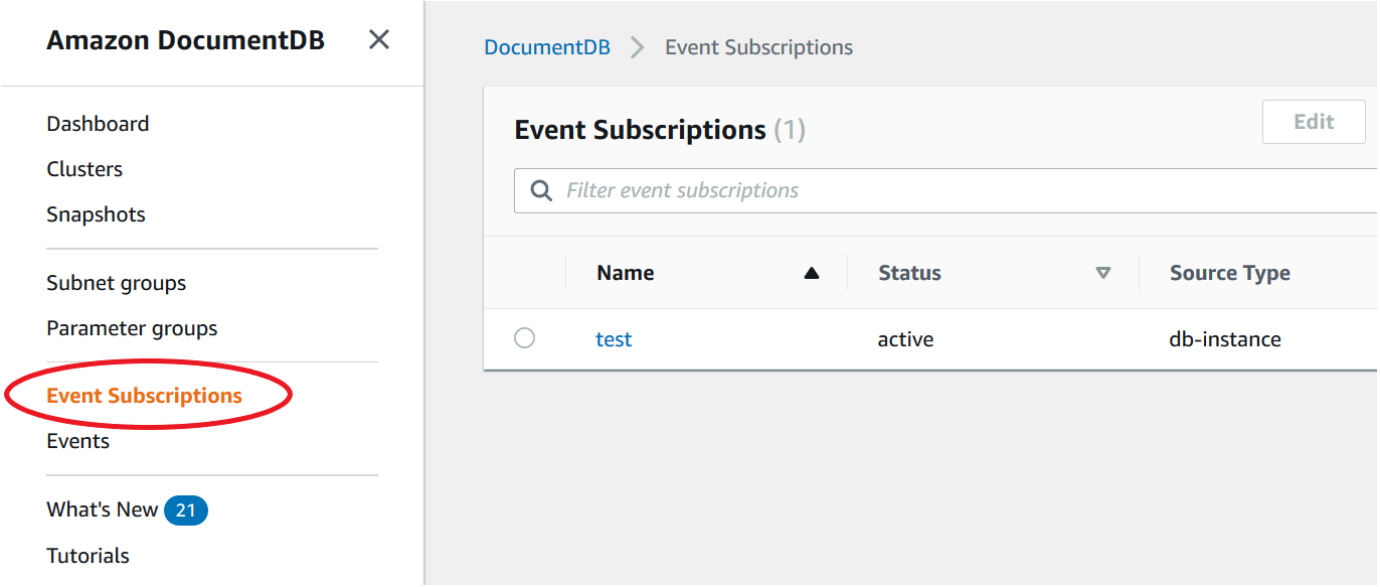
5. Wählen Sie Ändern aus. Die Amazon DocumentDB DocumentDB-Konsole zeigt an, dass das Abonnement geändert wird.



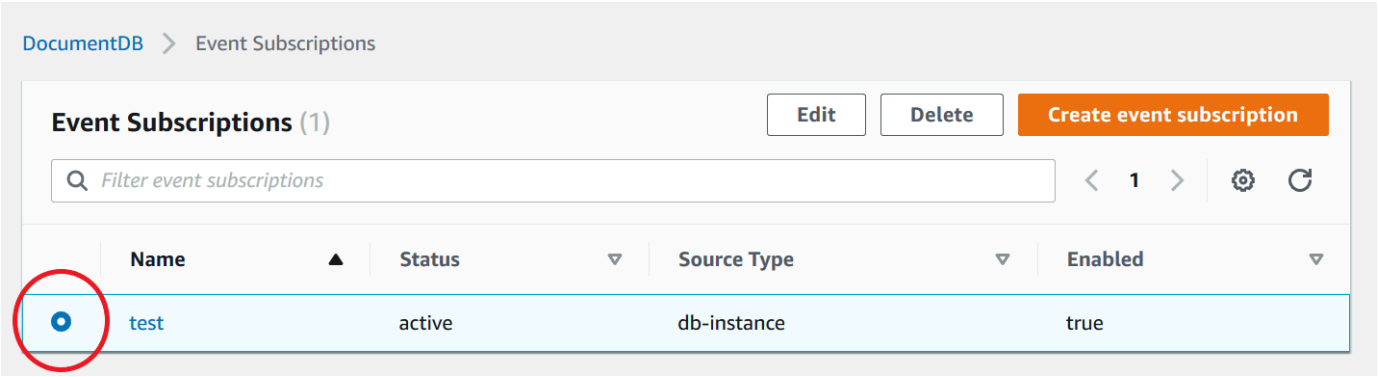
Löschen eines Abonnements für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen

Sie können ein Abonnement löschen, wenn Sie es nicht mehr benötigen. Alle Abonnenten des Themas erhalten dann keine weiteren Ereignisbenachrichtigungen, die über dieses Abonnement ausgegeben wurden.

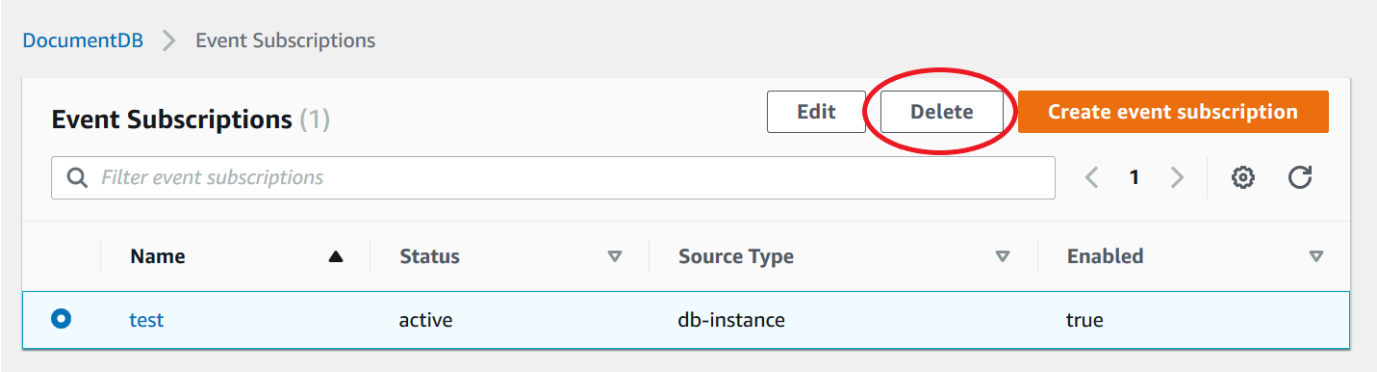
1. Melden Sie sich AWS Management Console unter an. https://console.aws.amazon.com/docdb
2. Wählen Sie im Navigationsbereich Ereignisabonnements aus.



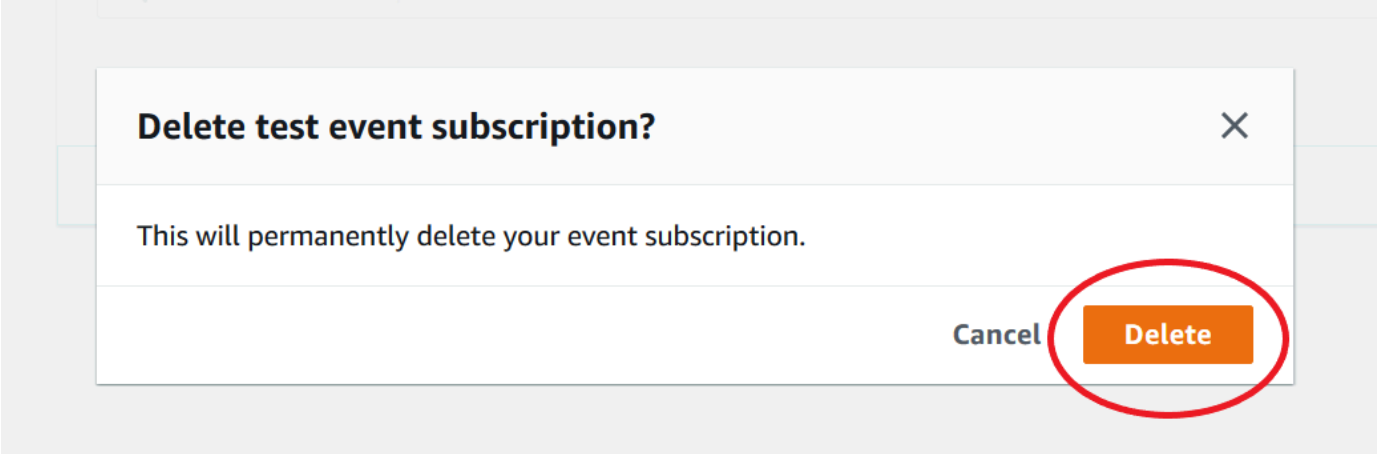
3. Wählen Sie im Bereich Ereignisabonnements das Abonnement aus, das Sie löschen möchten.



4. Wählen Sie Löschen.



5. In einem Popup-Fenster werden Sie gefragt, ob Sie diese Benachrichtigung dauerhaft löschen möchten. Wählen Sie Löschen.



Amazon DocumentDB DocumentDB-Ereigniskategorien und Meldungen

Amazon DocumentDB generiert eine beträchtliche Anzahl von Ereignissen in Kategorien, die Sie über die Konsole abonnieren können. Jede Kategorie bezieht sich auf einen Quelltyp, bei dem es sich um eine Instance, einen Cluster, einen Snapshot oder eine Parametergruppe handeln kann.

> **Note**
>
> Amazon DocumentDB verwendet bestehende Amazon RDS-Ereignisdefinitionen und IDs.

Amazon DocumentDB DocumentDB-Ereignisse, die von Instances stammen

Kategorie	Beschreibung
Verfügbarkeit	Die Instanz wurde neu gestartet.
Verfügbarkeit	Die Instanz wurde heruntergefahren.
Konfigurationsänderung	Änderung auf eine Instanzklasse anwenden.
Konfigurationsänderung	Das Anwenden der Änderung auf eine Instanzklasse wurde abgeschlossen.
Konfigurationsänderung	Primäre Anmeldeinformationen zurücksetzen.
Erstellung	Instanz wurde erstellt.
Löschung	Instanz wurde gelöscht
Ausfall	Die Instanz ist aufgrund einer inkompatiblen Konfiguration oder eines zugrunde liegenden Speicherproblems ausgefallen. Starten Sie a point-in-time-restore für die Instanz.
Benachrichtigung	Die Instanz wurde gestoppt.
Benachrichtigung	Instanz wurde gestartet.
Benachrichtigung	Die Instanz wird gestartet, weil sie die maximal zulässige Zeit für das Stoppen überschritten hat.
Wiederherstellung	Die Wiederherstellung der Instanz wurde gestartet. Die Wiederherstellungsdauer variiert je nach zu wiederherstellender Datenmenge.
Wiederherstellung	Die Wiederherstellung der Instanz ist abgeschlossen.

Kategorie	Beschreibung
Ausführen von Sicherheits-Patches	Das Betriebssystem-Update ist für Ihre Instance verfügbar. Informationen zum Anwenden von Updates finden Sie unter Wartung von Amazon DocumentDB.

Amazon DocumentDB DocumentDB-Ereignisse, die aus einem Cluster stammen

Kategorie	Beschreibung
Erstellung	Cluster wurde erstellt
Löschung	Cluster wurde gelöscht.
Failover	Der vorherige Primärserver wird erneut beworben.
Failover	Der Failover zur Instanz wurde abgeschlossen.
Failover	Der Failover zur DB-Instance wurde gestartet: %s
Failover	Derselbe AZ-Failover zur DB-Instance wurde gestartet: %s
Failover	Cross-AZ-Failover zur DB-Instance gestartet: %s
Wartung	Der Cluster wurde gepatcht.
Wartung	Der Datenbank-Cluster befindet sich in einem Zustand, der nicht aktualisiert werden kann: %s
Benachrichtigung	Der Cluster wurde gestoppt.
Benachrichtigung	Der Cluster wurde gestartet.
Benachrichtigung	Der Clusterstopp ist fehlgeschlagen.

Kategorie	Beschreibung
Benachrichtigung	Der Cluster wird gestartet, weil er die maximal zulässige Zeit für das Stoppen überschritten hat.
Benachrichtigung	Der Cluster wurde von %s in %s umbenannt.

Amazon DocumentDB DocumentDB-Ereignisse, die aus einem Cluster-Snapshot stammen

Die folgende Tabelle zeigt die Ereigniskategorie und eine Liste der Ereignisse, wenn ein Amazon DocumentDB-Cluster-Snapshot der Quelltyp ist.

Kategorie	Beschreibung
backup	Ein manueller Cluster-Snapshot wird erstellt.
backup	Ein manueller Cluster-Snapshot wurde erstellt.
backup	Erstellen eines automatisierten Cluster-S napshots.
backup	Automatisierter Cluster-Snapshot erstellt.

Amazon DocumentDB DocumentDB-Ereignisse, die aus einer Parametergruppe stammen

Die folgende Tabelle zeigt den Ereignistyp sowie eine Liste der Ereignisse für den Fall, dass der Quelltyp „Parametergruppe“ ist.

Kategorie	Beschreibung
Konfigurationsänderung	Der Parameter %s wurde mit der Methode %s auf %s aktualisiert

Überwachen von Amazon DocumentDB mit CloudWatch

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist in Amazon integriert, CloudWatch sodass Sie Betriebsmetriken für Ihre Cluster sammeln und analysieren können. Sie können diese Metriken mit der CloudWatch Konsole, der Amazon DocumentDB DocumentDB-Konsole, der AWS Command Line Interface (AWS CLI) oder der CloudWatch API überwachen.

CloudWatch ermöglicht es Ihnen auch, Alarme einzustellen, sodass Sie benachrichtigt werden können, wenn ein Metrikwert einen von Ihnen angegebenen Schwellenwert überschreitet.
Sie können Amazon CloudWatch Events sogar so einrichten, dass im Falle eines Verstoßes Korrekturmaßnahmen ergriffen werden. Weitere Informationen zur Verwendung CloudWatch und zu Alarmen finden Sie in der CloudWatch Amazon-Dokumentation.

Themen

- Amazon DocumentDB-Metriken
- CloudWatch Daten anzeigen
- Abmessungen von Amazon DocumentDB
- Überwachung von Opcounter-Metriken
- Überwachung von Datenbankverbindungen

Amazon DocumentDB-Metriken

Um den Zustand und die Leistung Ihres Amazon DocumentDB-Clusters und Ihrer Amazon DocumentDB-Instances zu überwachen, können Sie die folgenden Metriken in der Amazon DocumentDB-Konsole einsehen.

> **Note**
>
> Die Metriken in den folgenden Tabellen gelten sowohl für instanzbasierte als auch für elastische Cluster.

Themen

- Metriken zur Ressourcenauslastung
- Latenzmetriken
- NVMe-unterstützte Instance-Metriken

- Betriebsmetriken
- Kennzahlen zum Durchsatz
- Systemmetriken
- Metriken für T3-Instanzen

Metriken zur Ressourcenauslastung

Metrik	Beschreibung	
`BackupRetentionPeriodStorageUsed`	Die Gesamtmenge des Backup-Speichers in Byte, der zur Unterstützung der point-in-time Wiederherstellungsfunktion innerhalb des Aufbewahrungsfensters von Amazon DocumentDB verwendet wird. Ist in dem von der `TotalBackupStorageBilled` -Metrik gemeldeten Gesamtwert enthalten. Wird für jeden Amazon DocumentDB-Cluster separat berechnet.	
`ChangeStreamLogSize`	Die Menge des Speichers , der vom Cluster zum Speichern des Änderungsstream-Protokolls in Megabyte verwendet wird. Dieser Wert ist eine Teilmenge des Gesamtspeichers für den Cluster (`VolumeBytesUsed`) und wirkt sich auf die Kosten des Clusters aus. Informationen zu Speicherpreisen finden Sie auf der Amazon DocumentD	

Metrik	Beschreibung	
	B DocumentDB-Produkt seite. Die Protokollgröße des Änderungsstreams ist vom Umfang der Änderunge n auf Ihrem Cluster und der Aufbewahrungsdauer des Änderungsstream-Pr otokoll abhängig. Weitere Informationen zu Änderungs streams finden Sie unter Change-Streams mit Amazon DocumentDB verwenden.	
`CPUUtilization`	Prozentsatz der CPU-Nutzung durch eine Instance.	
`DatabaseConnections`	Die Anzahl der offenen Verbindungen auf einer Instance, die im Abstand von 1 Minute hergestellt wurde.	
`DatabaseConnection sMax`	Die maximale Anzahl offener Datenbankverbindungen auf einer Instance in einem Zeitraum von 1 Minute.	
`DatabaseCursors`	Die Anzahl der auf einer Instance geöffneten Cursor, die mit einer Frequenz von 1 Minute geöffnet werden.	
`DatabaseCursorsMax`	Die maximale Anzahl geöffneter Cursor auf einer Instance in einem Zeitraum von 1 Minute.	

Metrik	Beschreibung
DatabaseCursorsTimedOut	Die Anzahl der Cursor, bei denen innerhalb eines Zeitraums von 1 Minute ein Timeout aufgetreten ist.
FreeableMemory	Die Menge des verfügbaren Arbeitsspeichers (RAM-Speicher) in Bytes.
FreeLocalStorage	Diese Metrik gibt die Menge an Speicher an, der für jede Instance für temporäre Tabellen und Protokolle zur Verfügung steht. Dieser Wert hängt von der Instance-Klasse ab. Sie können die Menge an freiem Speicherplatz für eine Instance erhöhen, indem Sie eine größere Instance-Klasse für Ihre Instance auswählen.
LowMemThrottleQueueDepth	Die Warteschlangentiefe für Anfragen, die aufgrund des geringen verfügbaren Speichers gedrosselt werden, und werden in einer Frequenz von 1 Minute abgerufen.
LowMemThrottleMaxQueueDepth	Die maximale Warteschlangentiefe für Anfragen, die aufgrund von zu wenig verfügbarem Speicherplatz in einem Zeitraum von 1 Minute gedrosselt werden.

Metrik	Beschreibung
`LowMemNumOperationsThrottled`	Die Anzahl der Anfragen, die aufgrund von zu wenig verfügbarem Speicherplatz in einem Zeitraum von 1 Minute gedrosselt werden.
`SnapshotStorageUsed`	Die Gesamtmenge des Backup-Speichers in Byte, der von allen Snapshots für einen bestimmten Amazon DocumentDB-Cluster außerhalb seines Aufbewahrungsfensters für Backups verbraucht wird. Ist in dem von der `TotalBackupStorageBilled` -Metrik gemeldeten Gesamtwert enthalten. Wird für jeden Amazon DocumentDB-Cluster separat berechnet.
`SwapUsage`	Größe des auf der Instance genutzten Auslagerungsbereichs.

Metrik	Beschreibung
`TotalBackupStorageBilled`	Die Gesamtmenge an Backup-Speicher in Byte, die Ihnen für einen bestimmten Amazon DocumentDB-Cluster in Rechnung gestellt wird. Umfasst den Sicherungsspeicher gemessen an den Metriken `SnapshotStorageUsed` und `BackupRetentionPeriodStorageUsed`. Wird für jeden Amazon DocumentDB-Cluster separat berechnet.
`TransactionsOpen`	Die Anzahl der offenen Transaktionen auf einer Instance, die im Abstand von 1 Minute ausgeführt wurden.
`TransactionsOpenMax`	Die maximale Anzahl von Transaktionen, die in einem Zeitraum von 1 Minute auf einer Instance geöffnet werden.
`VolumeBytesUsed`	Die Menge des von Ihrem Cluster verwendeten Speicherplatzes in Byte. Dieser Wert wirkt sich auf die Kosten des Clusters aus. Preisinformationen finden Sie auf der Amazon DocumentDB DocumentDB-Produktseite.

Latenzmetriken

Metrik	Beschreibung
DBClusterReplicaLagMaximum	Die maximale Verzögerung in Millisekunden zwischen der primären Instance und jeder Amazon DocumentDB DocumentDB-Instance im Cluster.
DBClusterReplicaLagMinimum	Die Mindestverzögerung zwischen der Primär-Instance und jeder Replikat-Instance im Cluster, angegeben in Millisekunden.
DBInstanceReplicaLag	Die Verzögerung in Millisekunden, wenn Updates von der Primär-Instance an eine Replikat-Instance repliziert werden.
ReadLatency	Die durchschnittliche Dauer für einen Festplatten-E/A-Vorgang.
WriteLatency	Die durchschnittliche Dauer einer Datenträger-E/A-Operation in Millisekunden.

NVMe-unterstützte Instance-Metriken

Metrik	Beschreibung
`NVMeStorageCacheHitRatio`	Der Prozentsatz der Anfragen, die vom mehrstufigen Cache bedient werden.
`FreeNVMeStorage`	Die Menge des verfügbaren kurzlebigen Speichers NVMe .
`ReadIOPSNVMeStorage`	Die durchschnittliche Anzahl von I/O-Lesevorgängen auf der Festplatte für kurzlebigen NVMe Speicher.
`ReadLatencyNVMeStorage`	Die durchschnittliche Zeit, die pro I/O-Lesevorgang auf der Festplatte für NVMe Ephemeral Storage benötigt wird.
`ReadThroughputNVMeStorage`	Die durchschnittliche Anzahl der pro Sekunde von der Festplatte gelesenen Byte für kurzlebigen NVMe Speicher.
`WriteIOPSNVMeStorage`	Die durchschnittliche Anzahl von I/O-Schreibvorgängen auf der Festplatte in temporären Speicher. NVMe
`WriteLatencyNVMeStorage`	Die durchschnittliche Zeit, die pro I/O-Schreibvorgang auf der Festplatte für kurzlebigen NVMe Speicher benötigt wird.
`WriteThroughputNVMeStorage`	Die durchschnittliche Anzahl der pro Sekunde auf die

Metrik	Beschreibung	
	Festplatte geschriebenen Byte für kurzlebigen NVMe Speicher.	

Betriebsmetriken

Metrik	Beschreibung	
`DocumentsDeleted`	Die Anzahl der gelöschten Dokumente in einem Zeitraum von 1 Minute.	
`DocumentsInserted`	Die Anzahl der eingefügten Dokumente in einem Zeitraum von 1 Minute.	
`DocumentsReturned`	Die Anzahl der zurückgeg ebenen Dokumente in einem Zeitraum von 1 Minute.	
`DocumentsUpdated`	Die Anzahl der aktualisierten Dokumente in einem Zeitraum von 1 Minute.	
`OpcountersCommand`	Die Anzahl der Befehle, die in einem Zeitraum von 1 Minute ausgegeben wurden.	
`OpcountersDelete`	Die Anzahl der Löschvorg änge, die in einem Zeitraum von 1 Minute ausgeführt wurden.	
`OpcountersGetmore`	Die Anzahl der innerhalb einer Minute ausgegebenen Getmores.	

Metrik	Beschreibung
OpcountersInsert	Die Anzahl der innerhalb eines Zeitraums von 1 Minute ausgeführten Insert-Op erationen.
OpcountersQuery	Die Anzahl der Abfragen, die in einem Zeitraum von 1 Minute ausgegeben wurden.
OpcountersUpdate	Die Anzahl der Aktualisi erungsvorgänge, die in einem Zeitraum von 1 Minute ausgeführt wurden.
TransactionsStarted	Die Anzahl der Transaktionen, die in einem Zeitraum von 1 Minute auf einer Instance gestartet wurden.
TransactionsCommit ted	Die Anzahl der Transaktionen, die in einem Zeitraum von 1 Minute auf einer Instance festgeschrieben wurden.
TransactionsAborted	Die Anzahl der Transakti onen, die auf einer Instance in einem Zeitraum von 1 Minute abgebrochen wurden.
TTLDeletedDocuments	Die Anzahl der Dokumente , die von a TTLMonitor in einem Zeitraum von 1 Minute gelöscht wurden.

Kennzahlen zum Durchsatz

Metrik	Beschreibung
`NetworkReceiveThroughput`	Der von Clients erhaltene Netzwerkdurchsatz für jede Instance im Cluster, angegeben in Bytes pro Sekunde. Dieser Durchsatz beinhaltet nicht den Netzwerkdatenverkehr zwischen den Instances im Cluster und dem Cluster-Volume.
`NetworkThroughput`	Die Menge des Netzwerkdurchsatzes in Byte pro Sekunde, der von jeder Instance im Amazon DocumentDB-Cluster sowohl von Clients empfangen als auch an diese übertragen wird. Dieser Durchsatz beinhaltet nicht den Netzwerkdatenverkehr zwischen den Instances im Cluster und dem Cluster-Volume.
`NetworkTransmitThroughput`	Der an Clients gesendete Netzwerkdurchsatz für jede Instance im Cluster, angegeben in Bytes pro Sekunde. Dieser Durchsatz beinhaltet nicht den Netzwerkdatenverkehr zwischen den Instances im Cluster und dem Cluster-Volume.

Metrik	Beschreibung
ReadIOPS	Durchschnittliche Anzahl der Festplatten-I/O-Lesevorgänge pro Sekunde. Amazon DocumentDB meldet Lese- und Schreib-IOPS getrennt und in 1-Minuten-Intervallen.
ReadThroughput	Die durchschnittliche Anzahl Byte, die pro Sekunde vom Datenträger gelesen werden.
StorageNetworkReceiveThroughput	Die Menge des Netzwerkdurchsatzes in Byte pro Sekunde, den jede Instance im Cluster vom Amazon DocumentDB-Cluster-Speichervolume empfängt.
StorageNetworkTransmitThroughput	Die Menge an Netzwerkdurchsatz in Byte pro Sekunde, die von jeder Instance im Cluster an das Amazon DocumentDB-Cluster-Speichervolume gesendet wird.
StorageNetworkThroughput	Die Menge des Netzwerkdurchsatzes in Byte pro Sekunde, der von jeder Instance im Amazon DocumentDB-Cluster empfangen und an das Amazon DocumentDB-Cluster-Speichervolume gesendet wurde.

Metrik	Beschreibung	
VolumeReadIOPs	Die durchschnittliche Anzahl von in Rechnung gestellten I/O-Operationen aus einem Cluster-Volumen, meldet Werte in einem 5-Minuten -Intervall. In Rechnung gestellte Lesevorgänge werden auf Cluster-Volume-Ebene berechnet, aus allen Instances im Cluster zusammengestellt und dann in fünfminütigen Intervall en gemeldet. Der Wert wird anhand des Werts der Metrik für Leseoperationen über einen fünfminütigen Zeitraum berechnet. Sie können die Menge an in Rechnung gestellten Leseoperationen pro Sekunde bestimmen, indem Sie den Wert der Metrik für in Rechnung gestellte Leseoperationen durch 300 Sekunden teilen. Wenn der beispielsweise 13.686 VolumeReadIOPs zurückgibt, dann sind die abgerechneten Lesevorgänge pro Sekunde 45 (13.686/300 = 45,62). In Rechnung gestellte Operationen fallen für Abfragen für nicht im Buffer-	

Metrik	Beschreibung	
	Cache enthaltene Datenbankseiten an, die erst aus dem Speicher geladen werden müssen. Sie sehen evtl. Spitzenwerte in den in Rechnung gestellten Operationen, da Abfrageergebnisse aus dem Speicher gelesen und anschließend in den Buffer-Cache geladen werden.	

Metrik	Beschreibung	
VolumeWriteIOPs	Die durchschnittliche Anzahl von in Rechnung gestellte n Schreib-E/A-Operationen aus einem Cluster-Volume, gemeldet in 5-Minuten-Interval len. In Rechnung gestellte Schreibvorgänge werden auf Cluster-Volume-Ebene berechnet, aus allen Instances im Cluster aggregiert und dann in 5-Minuten-Interval len gemeldet. Der Wert wird anhand des Werts der Schreiboperationsmetrik über einen Zeitraum von 5 Minuten berechnet. Sie können die Menge der in Rechnung gestellten Schreiboperationen pro Sekunde ermitteln, indem Sie den Wert der Schreibop erations-Abrechnungsmetrik durch 300 Sekunden teilen. Wenn der beispielsweise 13.686 VolumeWriteIOPs zurückgibt, dann sind die fakturierten Schreibvorgänge pro Sekunde 45 (13.686/300 = 45,62). Beachten Sie, dass die VolumeReadIOPs VolumeWriteIOPs Metriken von der DocumentD B-Speicherebene berechnet	

Metrik	Beschreibung	
	werden und die von der Primär- und Replikatinstanz IOs ausgeführten Daten einschließen. Die Daten werden alle 20 bis 30 Minuten aggregiert und dann in Intervallen von 5 Minuten gemeldet, sodass in dem Zeitraum derselbe Datenpunkt für die Metrik ausgegeben wird. Wenn Sie nach einer Metrik suchen, die mit Ihren Einfügevorgängen über ein Intervall von 1 Minute korreliert, können Sie die WriteIOps -Metrik auf Instanzebene verwenden. Die Metrik ist auf der Registerkarte Überwachung Ihrer primären Amazon DocumentDB DocumentDB-Instance verfügbar.	
`WriteIOPS`	Durchschnittliche Anzahl von Festplatten-I/O-Schreibvorgänge pro Sekunde. Bei Verwendung auf Clustereb ene `WriteIOPs` werden sie für alle Instances im Cluster ausgewertet. Lese- und Schreib-IOPS werden separat und in 1-Minuten-Intervallen angegeben.	

Metrik	Beschreibung
`WriteThroughput`	Die durchschnittliche Anzahl von Bytes, die pro Sekunde auf den Datenträger geschrieben werden.

Systemmetriken

Metrik	Beschreibung
`BufferCacheHitRatio`	Der Prozentsatz der vom Buffer-Cache bedienten Anfragen.
`DiskQueueDepth`	Die Anzahl der I/O-Operationen, die darauf warten, auf die Festplatte geschrieben oder von ihr gelesen zu werden.
`EngineUptime`	Die Gesamtlaufzeit der Instance in Sekunden.
`IndexBufferCacheHitRatio`	Der Prozentsatz der Indexanforderungen, die vom Puffercache bedient werden. Möglicherweise stellen Sie unmittelbar nach dem Löschen eines Indexes, einer Sammlung oder einer Datenbank einen Anstieg von mehr als 100 Prozent für die Metrik fest. Dies wird nach 60 Sekunden automatisch korrigiert. Diese Einschrän

Metrik	Beschreibung	
	kung wird in einem future Patch-Update behoben.	

Metriken für T3-Instanzen

Metrik	Beschreibung	
`CPUCreditUsage`	Die Anzahl der CPU-Credits, die während des Messzeitr aums ausgegeben wurden.	
`CPUCreditBalance`	Die Anzahl der CPU-Credits, die eine Instance gesammelt hat. Dieser Saldo wird während der Steigerung der CPU-Leistung aufgebrau cht, da die CPU-Guthaben schneller verbraucht als erworben werden.	
`CPUSurplusCreditBa lance`	Die Anzahl der überschüs sigen CPU-Guthaben, die zur Aufrechterhaltung der CPU-Leistung ausgegeben wurden, wenn der CPUCredit Balance-Wert Null ist.	
`CPUSurplusCreditsC harged`	Die Anzahl der überschüs sigen CPU-Guthaben, die die maximale Anzahl an CPU-Credits überschre iten, die innerhalb von 24 Stunden verdient werden können, sodass zusätzliche Gebühren anfallen. Weitere	

Metrik	Beschreibung	
	Informationen finden Sie unter Überwachen Ihres CPU-Guthabens.	

CloudWatch Daten anzeigen

Sie können CloudWatch Amazon-Daten mit der CloudWatch Konsole, der Amazon DocumentDB-Konsole AWS Command Line Interface (AWS CLI) oder der CloudWatch API anzeigen.

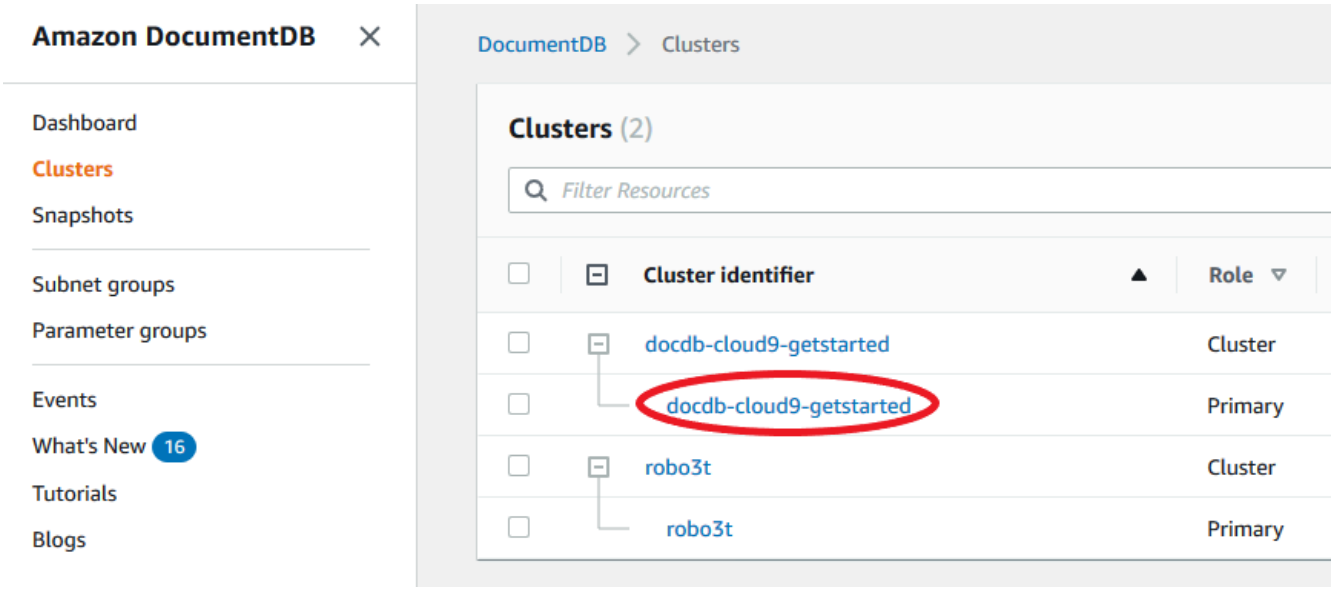
Using the AWS Management Console

Gehen Sie wie folgt vor, um CloudWatch Metriken mit der Amazon DocumentDB Management Console anzuzeigen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Klicken Sie im Navigationsbereich auf Cluster.

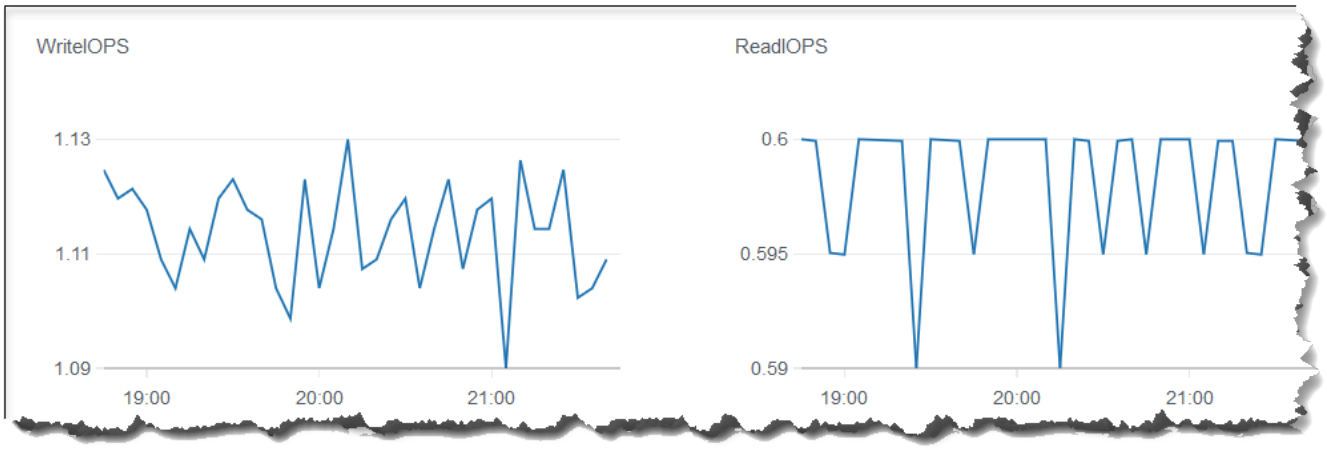
   > **Tip**
   >
   > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol (≡) aus.

3. Im Cluster-Navigationsfeld sehen Sie die Spalte Cluster-ID. Ihre Instances werden unter Clustern aufgeführt, ähnlich wie in der Abbildung unten.



4. Wählen Sie aus der Liste der Instances den Namen der Instance aus, für die Sie Metriken benötigen.
5. Wählen Sie auf der daraufhin angezeigten Instance-Übersichtsseite den Tab Monitoring, um grafische Darstellungen der Metriken Ihrer Amazon DocumentDB DocumentDB-Instance anzuzeigen. Da für jede Metrik ein Diagramm generiert werden muss, kann es einige Minuten dauern, bis die CloudWatchGrafiken gefüllt sind.

   Die folgende Abbildung zeigt die grafische Darstellung von zwei CloudWatch Metriken in der Amazon DocumentDB DocumentDB-Konsole, `WriteIOPS` und`ReadIOPS`.



Using the CloudWatch Management Console

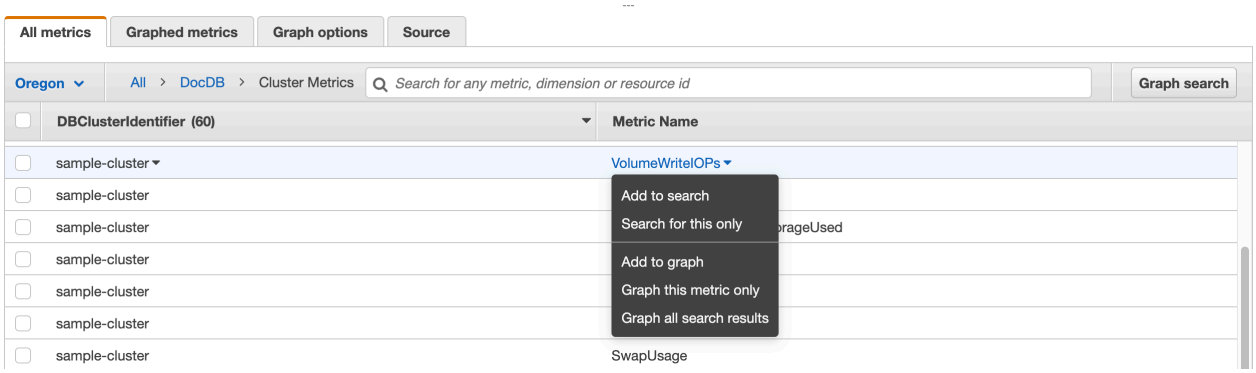
Gehen Sie wie folgt vor, um CloudWatch Metriken mit der CloudWatch Management Console anzuzeigen.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unterhttps://console.aws.amazon.com/cloudwatch.
2. Wählen Sie im Navigationsbereich Metriken aus. Wählen Sie dann aus der Liste der Dienstnamen DocDB aus.
3. Wählen Sie eine Metrikdimension (z. B. Cluster-Metriken).
4. Auf der Registerkarte Alle Metriken werden alle Metriken für diese Dimension in DocDB angezeigt.

    a. Um die Tabelle sortieren, verwenden Sie die Spaltenüberschrift.

    b. Um eine Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren. Um alle Metriken auszuwählen, aktivieren Sie das Kontrollkästchen in der Kopfzeile der Tabelle.

    c. Um nach Metrik zu filtern, bewegen Sie den Mauszeiger über den Metriknamen und wählen Sie den Dropdown-Pfeil neben dem Metriknamen aus. Wählen Sie dann Zur Suche hinzufügen aus, wie in der Abbildung unten gezeigt.



Using the AWS CLI

Verwenden Sie den CloudWatch `get-metric-statistics` Vorgang mit den folgenden Parametern, um CloudWatch Daten für Amazon DocumentDB anzuzeigen.

Parameter

- **`--namespace`** – Erforderlich. Der Service-Namespace, für den Sie CloudWatch -Metriken anzeigen möchten. Für Amazon DocumentDB muss dies der Fall sein. `AWS/DocDB`
- **`--metric-name`** – Erforderlich. Der Name der Metrik, für die Sie Daten benötigen.
- **`--start-time`** – Erforderlich. Der Zeitstempel, der den ersten zurückzugebenden Datenpunkt bestimmt.

Der angegebene Wert wird eingeschlossen. Die Ergebnisse umfassen Datenpunkte mit dem angegebenen Zeitstempel. Der Zeitstempel muss im ISO 8601 UTC-Format angegeben werden (z. B. 2016-10-03T23:00:00Z).

- **`--end-time`** – Erforderlich. Der Zeitstempel, der den letzten zurückzugebenden Datenpunkt bestimmt.

  Der angegebene Wert wird eingeschlossen. Die Ergebnisse umfassen Datenpunkte mit dem angegebenen Zeitstempel. Der Zeitstempel muss im ISO 8601 UTC-Format angegeben werden (z. B. 2016-10-03T23:00:00Z).

- **`--period`** – Erforderlich. Die Granularität der zurückgegebenen Datenpunkte in Sekunden. Für Metriken mit regulärer Auflösung kann ein Zeitraum gerade mal eine Minute (60 Sekunden) betragen. Der Wert muss ein Vielfaches von 60 sein. Für hochauflösende Metriken, die in Abständen von weniger als einer Minute erfasst werden, kann der Zeitraum 1, 5, 10, 30, 60 oder ein Vielfaches von 60 betragen.

- **`--dimensions`**— Fakultativ. Wenn die Metrik mehrere Dimensionen enthält, müssen Sie für jede Dimension einen Wert angeben. CloudWatch behandelt jede eindeutige Kombination von Dimensionen als separate Metrik. Wenn eine bestimmte Kombination von Dimensionen nicht veröffentlicht wurde, können Sie keine Metriken dafür abrufen. Sie müssen die gleichen Dimensionen angeben, die bei der Erstellung der Metriken verwendet wurden.

- **`--statistics`**— Fakultativ. Die Metrik-Statistiken abgesehen von den Perzentil-Statistiken. Verwenden Sie für Perzentil-Statistiken `ExtendedStatistics`. Wenn Sie `GetMetricStatistics` aufrufen, müssen Sie `Statistics` oder `ExtendedStatistics` angeben, aber nicht beides.

  Gültige Werte:

  - `SampleCount`
  - `Average`
  - `Sum`
  - `Minimum`
  - `Maximum`

- **`--extended-statistics`**— Fakultativ. Die `percentile`-Statistiken. Geben Sie Werte zwischen p0.0 und p100 an. Wenn Sie `GetMetricStatistics` aufrufen, müssen Sie `Statistics` oder `ExtendedStatistics` angeben, aber nicht beides.

- **`--unit`**— Fakultativ. Die Einheit für eine bestimmte Metrik. Metriken können in mehreren Einheiten angegeben werden. Wenn Sie keine Einheit festlegen, werden alle Einheiten zurückgegeben. Wenn Sie nur eine Einheit angeben, die für die Metrik nicht unterstützt wird, erhalten Sie keine Ergebnisse für den Aufruf.

  Mögliche Werte:

  - `Seconds`
  - `Microseconds`
  - `Milliseconds`
  - `Bytes`
  - `Kilobytes`
  - `Megabytes`
  - `Gigabytes`
  - `Terabytes`
  - `Bits`
  - `Kilobytes`
  - `Megabits`
  - `Gigabits`
  - `Terabits`
  - `Percent`
  - `Count`
  - `Bytes/Second`
  - `Kilobytes/Second`
  - `Megabytes/Second`
  - `Gigabytes/Second`
  - `Terabytes/Second`
  - `Bits/Second`
  - `Kilobits/Second`
  - `Megabits/Second`
  - `Gigabits/Second`
  - `Terabits/Second`

- `Count/Second`
- `None`

Example

Im folgenden Beispiel wird die maximale `CPUUtilization` für einen Zeitraum von 2 Stunden gezeigt, wobei alle 60 Sekunden eine Messung erfolgt.

Für Linux, macOS oder Unix:

```
aws cloudwatch get-metric-statistics \
       --namespace AWS/DocDB \
       --dimensions \
           Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 \
       --metric-name CPUUtilization \
       --start-time 2019-02-11T05:00:00Z \
       --end-time 2019-02-11T07:00:00Z \
       --period 60 \
       --statistics Maximum
```

Für Windows:

```
aws cloudwatch get-metric-statistics ^
       --namespace AWS/DocDB ^
       --dimensions ^
           Name=DBInstanceIdentifier,Value=docdb-2019-01-09-23-55-38 ^
       --metric-name CPUUtilization ^
       --start-time 2019-02-11T05:00:00Z ^
       --end-time 2019-02-11T07:00:00Z ^
       --period 60 ^
       --statistics Maximum
```

Die Ausgabe dieser Operation sieht in etwa wie folgt aus.

```
{
       "Label": "CPUUtilization",
       "Datapoints": [
           {
               "Unit": "Percent",
               "Maximum": 4.49152542374361,
```

```
            "Timestamp": "2019-02-11T05:51:00Z"
        },
        {
            "Unit": "Percent",
            "Maximum": 4.25000000000485,
            "Timestamp": "2019-02-11T06:44:00Z"
        },

        ********* some output omitted for brevity *********

        {
            "Unit": "Percent",
            "Maximum": 4.33333333331878,
            "Timestamp": "2019-02-11T06:07:00Z"
        }
    ]
}
```

Abmessungen von Amazon DocumentDB

Die Metriken für Amazon DocumentDB werden anhand der Werte für das Konto oder den Vorgang qualifiziert. Sie können die CloudWatch Konsole verwenden, um Amazon DocumentDB DocumentDB-Daten abzurufen, die nach einer der Dimensionen in der folgenden Tabelle gefiltert wurden.

Dimension	Beschreibung
`DBClusterIdentifier`	Filtert die Daten, die Sie für einen bestimmten Amazon DocumentDB-Cluster anfordern.
`DBClusterIdentifier, Role`	Filtert die Daten, die Sie für einen bestimmte n Amazon DocumentDB-Cluster anfordern, und aggregiert die Metrik nach Instance-Rolle (WRITER/READER). Sie können beispiels weise Metriken für alle READER-Instances eines Clusters zusammenfassen.
`DBInstanceIdentifier`	Filtert die angeforderten Daten für eine spezifische Datenbank-Instance.

Überwachung von Opcounter-Metriken

Opcounter-Metriken haben einen Wert ungleich Null (normalerweise ~50) für inaktive Cluster. Dies liegt daran, dass Amazon DocumentDB regelmäßige Zustandsprüfungen, interne Abläufe und Aufgaben zur Erfassung von Kennzahlen durchführt.

Überwachung von Datenbankverbindungen

Wenn Sie die Anzahl der Verbindungen mithilfe von Datenbankmodul-Befehlen wie anzeigen`db.runCommand( { serverStatus: 1 })`, werden möglicherweise bis zu 10 Verbindungen mehr angezeigt, als Sie `DatabaseConnections` durchschauen CloudWatch. Dies liegt daran, dass Amazon DocumentDB regelmäßige Zustandsprüfungen und Aufgaben zur Erfassung von Kennzahlen durchführt, die nicht berücksichtigt werden. `DatabaseConnections` `DatabaseConnections`steht nur für vom Kunden initiierte Verbindungen.

Protokollieren Amazon DocumentDB DocumentDB-API-Aufrufen mit AWS CloudTrail

Amazon DocumentDB (mit MongoDB-Kompatibilität) ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen von Benutzern, Rollen oder einem AWS Service in Amazon DocumentDB (mit MongoDB-Kompatibilität) bereitstellt. CloudTrail erfasst alle AWS CLI API-Aufrufe für Amazon DocumentDB als Ereignisse, einschließlich Aufrufe von der Amazon DocumentDB-Konsole und von Codeaufrufen an das Amazon DocumentDB-SDK. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon DocumentDB. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem auf der CloudTrail Konsole im Event-Verlauf einsehen. Anhand der von CloudTrail gesammelten Informationen können Sie die Anfrage an Amazon DocumentDB (mit MongoDB-Kompatibilität), die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde und andere Details ermitteln.

> **Important**
>
> Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird. Amazon DocumentDB DocumentDB-Konsolen- und API-Aufrufe werden als Aufrufe der Amazon RDS-API protokolliert. AWS CLI

Weitere Informationen AWS CloudTrail finden Sie im AWS CloudTrail Benutzerhandbuch.

Amazon DocumentDB DocumentDB-Informationen in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn Aktivitäten in Amazon DocumentDB (mit MongoDB-Kompatibilität) auftreten, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen. AWS-Konto Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich Ereignissen für Amazon DocumentDB (mit MongoDB-Kompatibilität), 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 anlegen, gilt dieser für alle AWS-Regionen-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS -Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie 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

Jedes Event oder jeder Protokolleintrag enthält Informationen über den Ersteller der Anfrage. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Root- oder -Benutzeranmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Profilierung von Amazon DocumentDB DocumentDB-Vorgängen

Sie können den Profiler in Amazon DocumentDB (mit MongoDB-Kompatibilität) verwenden, um die Ausführungszeit und Details der Operationen zu protokollieren, die auf Ihrem Cluster ausgeführt wurden. Profiler ist für die Überwachung der langsamsten Operationen in Ihrem Cluster nützlich. So können Sie die Leistung einzelner Abfragen und die allgemeine Cluster-Leistung verbessern.

Standardmäßig ist die Profiler-Funktion deaktiviert. Wenn diese Option aktiviert ist, protokolliert der Profiler Vorgänge, die länger dauern als ein vom Kunden definierter Schwellenwert (z. B. 100 ms), in Amazon CloudWatch Logs. Zu den protokollierten Details gehören der Befehl, für den das Profil erstellt wird, die Uhrzeit, die Planübersicht und die Client-Metadaten. Nachdem die Vorgänge in CloudWatch Logs protokolliert wurden, können Sie Logs Insights verwenden, um Ihre Amazon DocumentDB CloudWatch DocumentDB-Profilerstellungsdaten zu analysieren, zu überwachen und zu archivieren. Häufige Abfragen sind in Abschnitt Häufig gestellte Fragen zu finden.

Wenn aktiviert, verbraucht der Profiler zusätzliche Ressourcen im Cluster. Sie sollten mit einem hohen Schwellenwert (z. B. 500 ms) beginnen und den Wert allmählich senken, um langsame Operationen zu identifizieren. Ein anfänglicher Schwellenwert von 50 ms kann für Anwendungen mit hohem Durchsatz zu Leistungsproblemen im Cluster führen. Der Profiler ist auf Cluster-Ebene aktiviert und funktioniert auf allen Instances und Datenbanken in einem Cluster. Amazon DocumentDB protokolliert Operationen nach bestem Wissen und Gewissen in Amazon CloudWatch Logs.

Amazon DocumentDB erhebt zwar keine zusätzlichen Gebühren für die Aktivierung des Profilers, Ihnen werden jedoch die Standardtarife für die Nutzung von Logs berechnet. CloudWatch Informationen zu den Preisen für CloudWatch Logs finden Sie unter CloudWatch Amazon-Preise.

Themen

- Unterstützte Vorgänge
- Einschränkungen
- Den Amazon DocumentDB DocumentDB-Profiler aktivieren
- Den Amazon DocumentDB DocumentDB-Profiler deaktivieren
- Export von Profiler-Protokollen deaktivieren
- Zugreifen auf Ihre Amazon DocumentDB DocumentDB-Profiler-Protokolle
- Häufig gestellte Fragen

Unterstützte Vorgänge

Amazon DocumentDB Profiler unterstützt die folgenden Operationen:

- `aggregate`
- `count`
- `delete`
- `distinct`
- `find` (OP_QUERY und Befehl)
- `findAndModify`
- `insert`
- `update`

Einschränkungen

Der Profiler für langsame Abfragen kann Profiler-Protokolle nur ausgeben, wenn die gesamte Ergebnismenge der Abfrage in einen Stapel passt und wenn die Ergebnismenge weniger als 16 MB (maximale BSON-Größe) beträgt. Ergebnissätze, die größer als 16 MB sind, werden automatisch in mehrere Batches aufgeteilt.

Die meisten Treiber oder Shells legen möglicherweise eine kleine Standardstapelgröße fest. Sie können die Batchgröße als Teil Ihrer Abfrage angeben. Für die Erfassung langsamer Abfrageprotokolle empfehlen wir eine Batchgröße, die die Größe Ihrer erwarteten Ergebnismenge übersteigt. Wenn Sie sich bezüglich der Größe der Ergebnismenge nicht sicher sind oder wenn sie variiert, können Sie die Batchgröße auch auf eine große Zahl festlegen (z. B. 100.000).

Die Verwendung einer größeren Batchgröße bedeutet jedoch, dass mehr Ergebnisse aus der Datenbank abgerufen werden müssen, bevor eine Antwort an den Client gesendet wird. Bei einigen Abfragen kann dies zu längeren Verzögerungen führen, bis Sie Ergebnisse erhalten. Wenn Sie nicht vorhaben, die gesamte Ergebnismenge zu verwenden, ist es möglich, dass Sie mehr I/Os für die Verarbeitung der Abfrage aufwenden und das Ergebnis dann verwerfen.

Den Amazon DocumentDB DocumentDB-Profiler aktivieren

Die Aktivierung des Profilers auf einem Cluster erfolgt in drei Schritten. Stellen Sie sicher, dass alle Schritte abgeschlossen sind, da andernfalls keine Profilerstellungsprotokolle an Logs gesendet

CloudWatch werden. Profiler wird auf Cluster-Ebene festgelegt und auf allen Datenbanken und Instances des Clusters ausgeführt.

So aktivieren Sie den Profiler in einem Cluster

1. Da Sie eine Standard-Cluster-Parametergruppe nicht ändern können, stellen Sie sicher, dass Sie über eine verfügbare benutzerdefinierte Cluster-Parametergruppe verfügen. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen.
2. Ändern Sie mithilfe einer verfügbaren benutzerdefinierten Cluster-Parametergruppe die folgenden Parameter: `profiler`, `profiler_threshold_ms` und `profiler_sampling_rate`. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen ändern.
3. Erstellen oder ändern Sie Ihren Cluster, um die benutzerdefinierte Cluster-Parametergruppe zu verwenden und den Export von `profiler` Protokollen in Logs zu CloudWatch ermöglichen.

In den folgenden Abschnitten wird gezeigt, wie Sie diese Schritte mithilfe von AWS Management Console und AWS Command Line Interface (AWS CLI) implementieren.

Using the AWS Management Console

1. Bevor Sie beginnen, erstellen Sie einen Amazon DocumentDB-Cluster und eine benutzerdefinierte Cluster-Parametergruppe, falls Sie noch keinen haben. Weitere Informationen erhalten Sie unter Amazon DocumentDB-Cluster-Parametergruppen erstellen und Einen Amazon DocumentDB-Cluster erstellen.
2. Ändern Sie die folgenden Parameter mithilfe einer verfügbaren benutzerdefinierten Cluster-Parametergruppe. Weitere Informationen finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen ändern.

   - `profiler`— Aktiviert oder deaktiviert die Erstellung von Abfrageprofilen. Zugelassene Werte sind `enabled` und `disabled`. Der Standardwert ist `disabled`. Um Profiling zu aktivieren, legen Sie den Wert auf `enabled` fest.
   - `profiler_threshold_ms`— Wenn auf gesetzt `profiler` istenabled, werden alle Befehle protokolliert, die länger dauern als `profiler_threshold_ms` angegeben. CloudWatch Zugelassene Werte sind `[50-INT_MAX]`. Der Standardwert ist `100`.
   - `profiler_sampling_rate`— Der Anteil der langsamen Operationen, für die ein Profil erstellt oder protokolliert werden sollte. Zugelassene Werte sind `[0.0-1.0]`. Der Standardwert ist `1.0`.

3. Ändern Sie Ihren Cluster so, dass er die benutzerdefinierte Cluster-Parametergruppe verwendet, und legen Sie fest, dass die Profiler-Protokollexporte auf Amazon CloudWatch veröffentlicht werden sollen.

   a. Wählen Sie im Navigationsbereich die Option Clusters (Cluster), um die benutzerdefinierte Parametergruppe zu einem Cluster hinzuzufügen.

   b. Wählen Sie die Schaltfläche links neben dem Namen des Clusters, dem Sie die Parametergruppe zuordnen möchten. Wählen Sie Actions (Aktionen) und dann Modify (Ändern) aus, um den Cluster zu ändern.

   c. Wählen Sie unter Cluster options (Cluster-Optionen) die benutzerdefinierte Parametergruppe aus dem obigen Schritt aus, um sie Ihrem Cluster hinzuzufügen.

   d. Wählen Sie unter Protokollexporte die Option Profiler-Protokolle aus, die auf Amazon CloudWatch veröffentlicht werden sollen.

   e. Wählen Sie Continue (Weiter) aus, um eine Übersicht Ihrer Änderungen anzuzeigen.

   f. Nachdem Sie Ihre Änderungen überprüft haben, können Sie diese sofort oder während des nächsten Wartungsfensters unter Scheduling of modifications (Planen von Änderungen) anwenden.

   g. Wählen Sie Modify cluster (Cluster ändern) aus, um den Cluster mit der neuen Parametergruppe zu aktualisieren.

Using the AWS CLI

Mit dem folgenden Verfahren wird der Profiler für alle unterstützten Operationen für den Cluster `sample-cluster` aktiviert.

1. Bevor Sie beginnen, stellen Sie sicher, dass Sie über eine benutzerdefinierte Clusterparametergruppe verfügen, indem Sie den folgenden Befehl ausführen und die Ausgabe für eine Clusterparametergruppe überprüfen, die nicht `default` im Namen und die Parametergruppenfamilie `docdb3.6` hat. Wenn Sie über keine nicht standardmäßige Clusterparametergruppe verfügen, siehe Amazon DocumentDB-Cluster-Parametergruppen erstellen.

   ```
   aws docdb describe-db-cluster-parameter-groups \
       --query 'DBClusterParameterGroups[*].
   [DBClusterParameterGroupName,DBParameterGroupFamily]'
   ```

In der folgenden Ausgabe erfüllt nur `sample-parameter-group` beide Kriterien.

```
[
       [
           "default.docdb3.6",
           "docdb3.6"
       ],
       [
           "sample-parameter-group",
           "docdb3.6"
       ]
]
```

2. Ändern Sie mithilfe Ihrer benutzerdefinierten Cluster-Parametergruppe die folgenden Parameter:

   - `profiler`— Aktiviert oder deaktiviert die Erstellung von Abfrageprofilen. Zugelassene Werte sind `enabled` und `disabled`. Der Standardwert ist `disabled`. Um Profiling zu aktivieren, legen Sie den Wert auf `enabled` fest.
   - `profiler_threshold_ms`— Wenn auf gesetzt `profiler` ist`enabled`, dauern alle Befehle länger als protokolliert `profiler_threshold_ms` wurden. CloudWatch Zugelassene Werte sind `[50-INT_MAX]`. Der Standardwert ist `100`.
   - `profiler_sampling_rate`— Der Anteil langsamer Operationen, für den ein Profil erstellt oder protokolliert werden sollte. Zugelassene Werte sind `[0.0-1.0]`. Der Standardwert ist `1.0`.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=enabled,ApplyMethod=immediate \

 ParameterName=profiler_threshold_ms,ParameterValue=100,ApplyMethod=immediate \

 ParameterName=profiler_sampling_rate,ParameterValue=0.5,ApplyMethod=immediate
```

3. Ändern Sie Ihren Amazon DocumentDB-Cluster so, dass er die `sample-parameter-group` benutzerdefinierte Cluster-Parametergruppe aus dem vorherigen Schritt verwendet und den Parameter `--enable-cloudwatch-logs-exports` auf `profiler` setzt.

Der folgende Code ändert den Cluster so`sample-cluster`, dass er den `sample-parameter-group` aus dem vorherigen Schritt verwendet, und erweitert `profiler` die aktivierten CloudWatch Logs-Exporte.

```
aws docdb modify-db-cluster \
       --db-cluster-identifier sample-cluster \
       --db-cluster-parameter-group-name sample-parameter-group \
       --cloudwatch-logs-export-configuration '{"EnableLogTypes":["profiler"]}'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2020-04-07T02:05:12.479Z",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "LatestRestorableTime": "2020-04-08T22:08:59.317Z",
        "Port": 27017,
        "MasterUsername": "test",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
```

```
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-
cluster",
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "EnabledCloudwatchLogsExports": [
            "profiler"
        ],
        "DeletionProtection": true
    }
}
```

Den Amazon DocumentDB DocumentDB-Profiler deaktivieren

Um den Profiler zu deaktivieren, deaktivieren Sie sowohl den `profiler` Parameter als auch den Export von `profiler` Protokollen in Logs. CloudWatch

Deaktivieren des Profilers

Sie können den `profiler` Parameter entweder mit AWS Management Console oder AWS CLI wie folgt deaktivieren.

Using the AWS Management Console

Das folgende Verfahren verwendet die AWS Management Console , um Amazon DocumentDB `profiler` zu deaktivieren.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus. Wählen Sie dann den Namen der Cluster-Parametergruppe aus, für die Sie den Profiler deaktivieren möchten.
3. Klicken Sie auf der Seite Cluster parameters (Clusterparameter) auf die Schaltfläche links neben dem `profiler`-Parameter und wählen Sie Edit (Bearbeiten).
4. Wählen Sie im Dialogfeld Modify profiler (Profiler ändern) `disabled` in der Liste aus.
5. Wählen Sie Modify Cluster Parameter (Cluster-Parameter ändern).

Using the AWS CLI

Um `profiler` auf einem Cluster mithilfe der AWS CLI zu deaktivieren, ändern Sie den Cluster wie folgt.

```
aws docdb modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group \
    --parameters
 ParameterName=profiler,ParameterValue=disabled,ApplyMethod=immediate
```

Export von Profiler-Protokollen deaktivieren

Sie können den Export von `profiler` Protokollen in Logs deaktivieren, CloudWatch indem Sie entweder das AWS Management Console oder verwenden AWS CLI, und zwar wie folgt.

Using the AWS Management Console

Das folgende Verfahren verwendet die AWS Management Console , um den Export von Protokollen durch Amazon DocumentDB zu CloudWatch deaktivieren.

1. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

2. Klicken Sie im Navigationsbereich auf Cluster. Wählen Sie die Schaltfläche links neben dem Namen des Clusters, für den Sie den Export von Protokollen deaktivieren möchten.
3. Wählen Sie aus dem Menü Actions (Aktionen) die Option Modify (Ändern) aus.
4. Scrollen Sie nach unten zum Abschnitt Log exports (Protokollexporte) und deaktivieren Sie Profiler logs (Profiler-Protokolle).
5. Klicken Sie auf Weiter.
6. Überprüfen Sie Ihre Änderungen und wählen Sie dann aus, wann diese Änderung auf Ihren Cluster angewendet werden soll.
   - Apply during the next scheduled maintenance window (Anwendung während des nächsten geplanten Wartungsfensters)
   - Apply immediately (Sofort anwenden)
7. Wählen Sie Cluster bearbeiten aus.

Using the AWS CLI

Der folgende Code modifiziert den Cluster `sample-cluster` und deaktiviert Profiler-Logs. CloudWatch

Example

Für Linux, macOS oder Unix:

```
aws docdb modify-db-cluster \
   --db-cluster-identifier sample-cluster \
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Für Windows:

```
aws docdb modify-db-cluster ^
   --db-cluster-identifier sample-cluster ^
   --cloudwatch-logs-export-configuration '{"DisableLogTypes":["profiler"]}'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "DBCluster": {
        "AvailabilityZones": [
            "us-east-1c",
```

```
            "us-east-1b",
            "us-east-1a"
        ],
        "BackupRetentionPeriod": 1,
        "DBClusterIdentifier": "sample-cluster",
        "DBClusterParameterGroup": "sample-parameter-group",
        "DBSubnetGroup": "default",
        "Status": "available",
        "EarliestRestorableTime": "2020-04-08T02:05:17.266Z",
        "Endpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "ReaderEndpoint": "sample-cluster.node.us-east-1.docdb.amazonaws.com",
        "MultiAZ": false,
        "Engine": "docdb",
        "EngineVersion": "3.6.0",
        "LatestRestorableTime": "2020-04-09T05:14:44.356Z",
        "Port": 27017,
        "MasterUsername": "test",
        "PreferredBackupWindow": "02:00-02:30",
        "PreferredMaintenanceWindow": "tue:09:50-tue:10:20",
        "DBClusterMembers": [
            {
                "DBInstanceIdentifier": "sample-instance-1",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            },
            {
                "DBInstanceIdentifier": "sample-instance-2",
                "IsClusterWriter": true,
                "DBClusterParameterGroupStatus": "in-sync",
                "PromotionTier": 1
            }
        ],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-abcd0123",
                "Status": "active"
            }
        ],
        "HostedZoneId": "ABCDEFGHIJKLM",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:<accountID>:key/sample-key",
        "DbClusterResourceId": "cluster-ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "DBClusterArn": "arn:aws:rds:us-east-1:<accountID>:cluster:sample-cluster",
```

```
        "AssociatedRoles": [],
        "ClusterCreateTime": "2020-01-10T22:13:38.261Z",
        "DeletionProtection": true
    }
}
```

Zugreifen auf Ihre Amazon DocumentDB DocumentDB-Profiler-Protokolle

Gehen Sie wie folgt vor, um auf Ihre Profilprotokolle bei Amazon zuzugreifen CloudWatch.

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Stellen Sie sicher, dass Sie sich in derselben Region wie Ihr Amazon DocumentDB-Cluster befinden.
3. Wählen Sie im Navigationsbereich Protokolle aus.
4. Um die Profiler-Protokolle für Ihren Cluster zu finden, wählen Sie in der Liste `/aws/docdb/`*`yourClusterName`*`/profiler` aus.

   Die Profilprotokolle für jede Ihrer Instances sind unter den jeweiligen Instance-Namen verfügbar.

Häufig gestellte Fragen

Im Folgenden finden Sie einige häufige Abfragen, die Sie verwenden können, um Ihre Befehle, für die Profile erstellt wurden, zu analysieren. Weitere Informationen zu CloudWatch Logs Insights finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights und Beispielabfragen.

Ruft die 10 langsamsten Vorgänge für eine angegebene Sammlung ab

```
filter ns="test.foo" | sort millis desc | limit 10
```

Ruft alle Aktualisierungsvorgänge für eine Sammlung ab, die mehr als 60 ms gedauert haben

```
filter millis > 60 and op = "update"
```

Holen Sie sich die 10 langsamsten Operationen des letzten Monats

```
sort millis desc | limit 10
```

Abrufen aller Abfragen mit einer COLLSCAN-Planübersicht

```
filter planSummary="COLLSCAN"
```

Überwachung mit Performance Insights

Performance Insights erweitert die bestehenden Amazon DocumentDB DocumentDB-Überwachungsfunktionen, um die Leistung Ihres Clusters zu veranschaulichen und Sie bei der Analyse aller Probleme zu unterstützen, die sich darauf auswirken. Mit dem Performance Insights Insights-Dashboard können Sie die Datenbanklast visualisieren und die Last nach Wartezeiten, Abfrageanweisungen, Hosts oder Anwendungen filtern.

> **Note**
>
> Performance Insights ist nur für instanzbasierte Amazon DocumentDB 3.6-, 4.0- und 5.0-Cluster verfügbar.

Wie ist es nützlich?

- Datenbankleistung visualisieren — Visualisieren Sie die Last, um festzustellen, wann und wo sich die Last in der Datenbank befindet
- Ermitteln Sie, was die Belastung der Datenbank verursacht — Ermitteln Sie, welche Abfragen, Hosts und Anwendungen zur Belastung Ihrer Instance beitragen
- Ermitteln Sie, wann Ihre Datenbank ausgelastet ist — Vergrößern Sie das Performance Insights Insights-Dashboard, um sich auf bestimmte Ereignisse zu konzentrieren, oder zoomen Sie heraus, um Trends über einen längeren Zeitraum zu betrachten
- Warnung beim Laden der Datenbank — Automatischer Zugriff auf neue Datenbank-Lademetriken, von CloudWatch wo aus Sie die DB-Lademetriken zusammen mit anderen Amazon DocumentDB-Metriken überwachen und Warnmeldungen zu diesen einrichten können

Was sind die Einschränkungen von Amazon DocumentDB Performance Insights?

- Performance Insights in den Regionen AWS GovCloud (USA-Ost) und AWS GovCloud (US-West) sind nicht verfügbar
- Performance Insights for Amazon DocumentDB speichert Leistungsdaten für bis zu 7 Tage
- Abfragen, die länger als 1.024 Byte sind, werden in Performance Insights nicht aggregiert

Themen

- Konzepte von Performance Insights
- Aktivieren und Deaktivieren von Performance Insights
- Konfigurieren von Zugriffsrichtlinien für Performance Insights
- Analyse der Metriken mit dem Performance Insights-Dashboard
- Abrufen von Metriken mit der Performance Insights-API
- CloudWatch Amazon-Metriken für Performance Insights
- Performance Insights für Zählermetriken

Konzepte von Performance Insights

Themen

- Durchschnittliche aktive Sitzungen
- Dimensionen
- Max. vCPU

Durchschnittliche aktive Sitzungen

Datenbanklast (DB-Last) misst den Aktivitätsgrad in Ihrer Datenbank. Die wichtigste Metrik für Performance Insights ist `DB Load`, die jede Sekunde erfasst wird. Die Einheit für die `DBLoad` Metrik ist der Average Active Sessions (AAS) für eine Amazon DocumentDB DocumentDB-Instance.

Eine aktive Sitzung ist eine Verbindung, die Arbeit an die Amazon DocumentDB DocumentDB-Instance übermittelt hat und auf eine Antwort wartet. Wenn Sie beispielsweise eine Anfrage an eine Amazon DocumentDB DocumentDB-Instance senden, ist die Datenbanksitzung aktiv, während die Instance die Abfrage verarbeitet.

Um die Anzahl der durchschnittlich aktiven Sitzungen AAS zu erhalten, ruft Performance Insights die Anzahl der Sitzungen ab, die gleichzeitig eine Abfrage ausführen. Die AAS ist die Gesamtzahl

der Sitzungen geteilt durch die Gesamtzahl der Beispiele. Die folgende Tabelle zeigt fünf aufeinanderfolgende Beispiele einer laufenden Abfrage.

Beispiel	Anzahl der Sitzungen, die eine Abfrage ausführen	AAS	Berechnung
1	2	2	2 Sitzungen/1 Beispiel
2	0	1	2 Sitzungen /2 Beispiele
3	4	2	6 Sitzungen /3 Beispiele
4	0	1.5	6 Sitzungen /4 Beispiele
5	4	2	10 Sitzungen /5 Beispiele

Im vorherigen Beispiel beträgt der DB-Load für das Zeitintervall von 1-5 2 AAS. Eine Erhöhung der DB-Last bedeutet, dass im Durchschnitt mehr Sitzungen für die Datenbank ausgeführt werden.

Dimensionen

Die`DB Load` Metrik unterscheidet sich von den anderen Zeitreihenmetriken, da Sie sie in Unterkomponenten aufteilen können, die als Dimensionen bezeichnet werden. Sie können sich Dimensionen als Kategorien für die verschiedenen Merkmale der `DB Load`-Metrik vorstellen. Bei der Diagnose von Leistungsproblemen sind die Dimensionen Wait States und Top Query am nützlichsten.

Wartezustände

Ein Wartestatus bewirkt, dass eine Abfrageanweisung auf das Eintreten eines bestimmten Ereignisses wartet, bevor sie weiter ausgeführt werden kann. Beispielsweise kann eine Abfrageanweisung warten, bis eine gesperrte Ressource entsperrt ist. Durch die Kombination `DB Load` mit Wartezuständen können Sie sich ein vollständiges Bild vom Sitzungsstatus machen. Hier sind verschiedene Wartezustände von Amazon DocumentDB:

Amazon DocumentDB DocumentDB-Wartestatus	Beschreibung des Wartestatus
Riegeln	Der Latch-Wartestatus tritt auf, wenn die Sitzung darauf wartet, den Pufferpool auszulagern. Häufiges Ein- und Auslagern in den Pufferpool kann häufiger vorkommen , wenn das System häufig umfangreiche Abfragen verarbeitet, Sammlungsscans durchführt oder wenn der Pufferpool zu klein ist, um den Arbeitssatz zu verarbeiten.
CPU	Der CPU-Wartestatus tritt auf, wenn die Sitzung auf die CPU wartet.
CollectionLock	Der CollectionLock Wartestatus tritt ein, wenn die Sitzung darauf wartet, eine Sperre für die Sammlung zu erlangen. Diese Ereignisse treten auf, wenn DDL-Operationen für die Sammlung ausgeführt werden.
DocumentLock	Der DocumentLock Wartestatus tritt ein, wenn die Sitzung darauf wartet, ein Dokument zu sperren. Eine hohe Anzahl gleichzeitiger Schreibvorgänge auf dasselbe Dokument führt zu mehr DocumentLock Wartezuständen in diesem Dokument.
SystemLock	Der SystemLock Wartestatus tritt auf, wenn die Sitzung auf dem System wartet. Dies kann der Fall sein, wenn häufig Abfragen mit langer Laufzeit, lang andauernde Transaktionen oder eine hohe Parallelität im System auftreten.
IO	Der IO-Wartestatus tritt auf, wenn die Sitzung auf den Abschluss der IO wartet.

Amazon DocumentDB DocumentDB-Wartestatus	Beschreibung des Wartestatus
BufferLock	Der BufferLock Wartestatus tritt ein, wenn die Sitzung darauf wartet, eine Sperre für eine gemeinsam genutzte Seite im Puffer zu erlangen. BufferLockWartezustände können verlängert werden, wenn andere Prozesse die Cursor auf den angeforderten Seiten offen halten.
LowMemThrottle	Der LowMemThrottle Wartestatus tritt auf, wenn die Sitzung aufgrund einer hohen Speicherauslastung auf der Amazon DocumentDB DocumentDB-Instance wartet. Wenn dieser Status für eine lange Zeit andauert, sollten Sie erwägen, die Instance zu skalieren, um zusätzlichen Speicher bereitzustellen. Weitere Informationen finden Sie unter Resource Governor.
BackgroundActivity	Der BackgroundActivity Wartestatus tritt auf, wenn die Sitzung auf interne Systemprozesse wartet.
Sonstige	Der Wartestatus Andere ist ein interner Wartestatus. Wenn dieser Status über einen längeren Zeitraum andauert, sollten Sie erwägen, diese Abfrage zu beenden. Weitere Informationen finden Sie unter Wie finde und beende ich Abfragen mit langer Laufzeit oder blockierter Ausführung?

Die häufigsten Abfragen

Während bei Wartezuständen Engpässe auftreten, zeigen Abfragen am häufigsten, welche Abfragen am meisten zur Datenbanklast beitragen. Beispielsweise könnten derzeit viele Abfragen gleichzeitig

in der Datenbank ausgeführt werden, aber eine einzelne Abfrage könnte 99 % der DB-Last verbrauchen. In diesem Fall könnte die hohe Belastung auf ein Problem mit der Abfrage hinweisen.

Max. vCPU

Das Diagramm Datenbanklast im Dashboard dient zum Erfassen, Aggregieren und Anzeigen von Sitzungsinformationen. Um zu sehen, ob aktive Sitzungen die maximale CPU überschreiten, sehen Sie sich ihre Beziehung zur Max vCPU-Linie an. Der Max vCPU-Wert wird durch die Anzahl der vCPU-Kerne (virtuelle CPU) für Ihre Amazon DocumentDB DocumentDB-Instance bestimmt.

Wenn die DB-Last häufig über der Max vCPU-Linie liegt und der primäre Wartezustand „CPU“ lautet, ist die CPU überlastet. In diesem Fall möchten Sie möglicherweise die Verbindungen zur Instance drosseln, alle Abfragen mit hoher CPU-Last optimieren oder eine größere Instance-Klasse in Betracht ziehen. Hohe und konsistente Instances von Wartezuständen deuten darauf hin, dass es möglicherweise Engpässe oder Probleme mit Ressourcenkonflikten gibt, die behoben werden müssen. Dies kann auch dann zutreffen, wenn die DB-Last die mit Max vCPU definierte Linie nicht überschreitet.

Aktivieren und Deaktivieren von Performance Insights

Um Performance Insights zu nutzen, aktivieren Sie es auf Ihrer DB-Instance. Sie können sie bei Bedarf später deaktivieren. Das Aktivieren und Deaktivieren von Performance Insights führt nicht zu Ausfallzeiten, einem Neustart oder einem Failover.

Der Performance Insights-Agent verbraucht eine begrenzte Menge an CPU und Arbeitsspeicher auf dem DB-Host. Wenn die DB-Last hoch ist, begrenzt der Agent die Auswirkungen auf die Leistung, indem Daten seltener erfasst werden.

Aktivieren von Performance Insights beim Erstellen eines Clusters

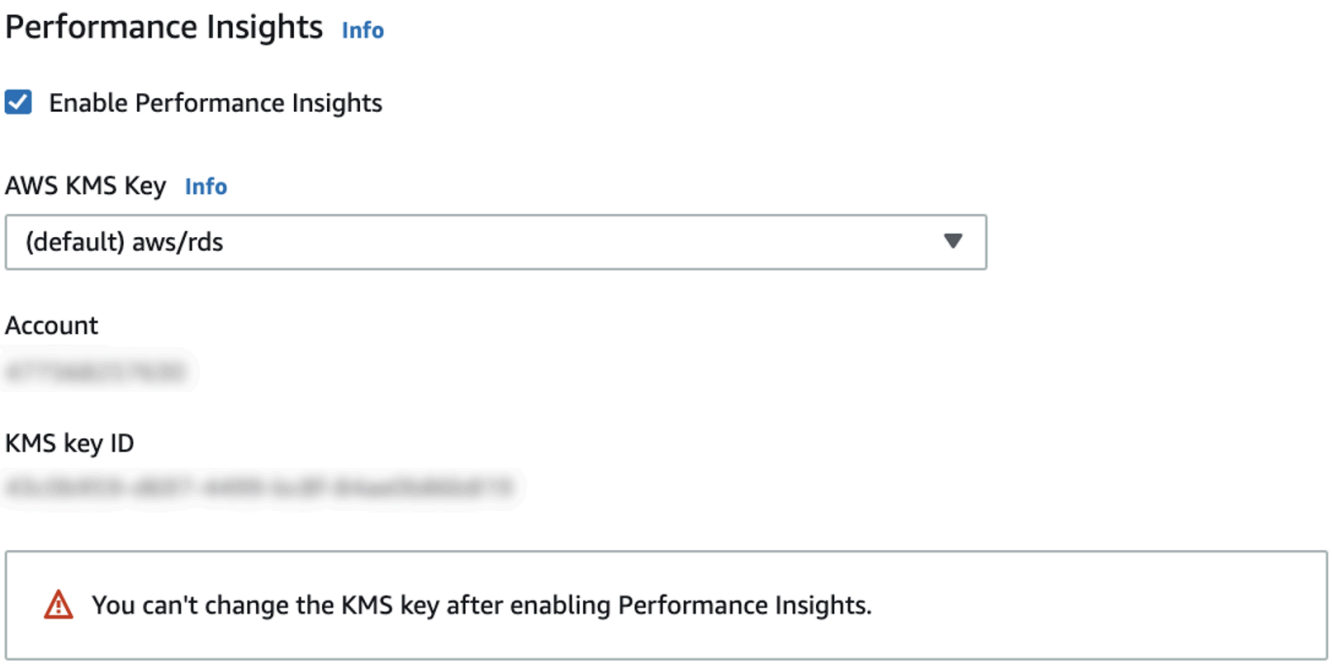
In der Konsole können Sie Performance Insights aktivieren oder deaktivieren, wenn Sie eine neue DB-Instance erstellen oder ändern.

Mit dem AWS Management Console

In der Konsole können Sie Performance Insights aktivieren, wenn Sie einen Amazon DocumentDB-Cluster erstellen. Wenn Sie einen neuen Amazon DocumentDB-Cluster erstellen, aktivieren Sie Performance Insights, indem Sie im Abschnitt Performance Insights die Option Performance Insights aktivieren wählen.

Anweisungen für die Konsole

1. Um einen Cluster zu erstellen, folgen Sie den Anweisungen unter Erstellen eines Amazon DocumentDB-Clusters.
2. Wählen Sie im Abschnitt Performance Insights die Option Performance Insights aktivieren aus.



> **Note**
>
> Die Aufbewahrungsfrist für Performance Insights Insights-Daten beträgt sieben Tage.

AWS KMS Schlüssel — Geben Sie Ihren AWS KMS-Schlüssel an. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem AWS KMS Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Konfiguration einer AWS AWS KMS Richtlinie für Performance Insights.

Aktivierung und Deaktivierung beim Ändern einer Instanz

Sie können eine DB-Instance ändern, um Performance Insights mithilfe der Konsole oder zu aktivieren oder zu deaktivieren AWS CLI.

Using the AWS Management Console

Anweisungen für die Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie Clusters (Cluster) aus.
3. Wählen Sie eine DB-Instance aus und wählen Sie Ändern.
4. Wählen Sie im Abschnitt Performance Insights entweder Performance Insights aktivieren oder Performance Insights deaktivieren aus.

   > **Note**
   >
   > Wenn Sie Enable Performance Insights wählen, können Sie Ihren AWS AWS KMS Schlüssel angeben. Performance Insights verschlüsselt alle potenziell sensiblen Daten mit Ihrem AWS KMS Schlüssel. Die Daten werden während der Übertragung und im Ruhezustand verschlüsselt. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Daten im Ruhezustand verschlüsseln.

5. Klicken Sie auf Weiter.
6. Wählen Sie für Einplanung von Änderungen die Option Sofort anwenden aus. Wenn Sie während des nächsten geplanten Wartungsfensters Anwenden wählen, ignoriert Ihre Instance diese Einstellung und aktiviert Performance Insights sofort.
7. Wählen Sie Modify instance (Instance ändern).

Using the AWS CLI

Wenn Sie die `modify-db-instance` AWS AWS CLI Befehle `create-db-instance` oder verwenden, können Sie Performance Insights durch Angabe aktivieren oder deaktivieren`--enable-performance-insights`, indem Sie angeben`--no-enable-performance-insights`.

Im folgenden Verfahren wird beschrieben, wie Sie mit der AWS AWS CLI Performance-Insights für eine DB-Instance aktivieren oder deaktivieren.

AWS AWS CLI Anweisungen

Rufen Sie den `modify-db-instance` AWS AWS CLI Befehl auf und geben Sie die folgenden Werte an:

- `--db-instance-identifer`— Der Name der DB-Instance
- `--enable-performance-insights` zum Aktivieren oder `--no-enable-performance-insights` zum Deaktivieren

Example

Das folgende Beispiel aktiviert Performance Insights für`sample-db-instance`:

For Linux, macOS, or Unix:

```
aws docdb modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights
```

For Windows:

```
aws docdb modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
    --enable-performance-insights
```

Konfigurieren von Zugriffsrichtlinien für Performance Insights

Um auf Performance Insights zugreifen zu können, müssen Sie über die entsprechenden Berechtigungen von AWS Identity and Access Management (IAM) verfügen. Sie haben folgende Möglichkeiten, Zugriff zu gewähren:

- Fügen Sie die verwaltete Richtlinie `AmazonRDSPerformanceInsightsReadOnly` an einen Berechtigungssatz oder eine Rolle an.
- Erstellen Sie eine benutzerdefinierte IAM-Richtlinie und fügen Sie diese an einen Berechtigungssatz oder eine Rolle an.

Wenn Sie bei der Aktivierung von Performance Insights einen vom Kunden verwalteten Schlüssel angegeben haben, stellen Sie außerdem sicher, dass die Benutzer in Ihrem Konto über die Berechtigungen `kms:Decrypt` und `kms:GenerateDataKey` für den KMS-Schlüssel verfügen.

> **Note**
>
> Für die encryption-at-rest Verwaltung von AWS KMS Schlüsseln und Sicherheitsgruppen nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS gemeinsam genutzt wird.

Anhängen der RDSPerformance InsightsReadOnly Amazon-Richtlinie an einen IAM-Principal

`AmazonRDSPerformanceInsightsReadOnly`ist eine AWS verwaltete Richtlinie, die Zugriff auf alle schreibgeschützten Operationen der Amazon DocumentDB Performance Insights-API gewährt. Derzeit sind alle Operationen in dieser API schreibgeschützt. Wenn Sie `AmazonRDSPerformanceInsightsReadOnly` an einen Berechtigungssatz oder eine Rolle anfügen, kann der Empfänger Performance Insights mit anderen Konsolenfunktionen verwenden.

Erstellen einer benutzerdefinierten IAM-Richtlinie für Performance Insights

Für Benutzer, die nicht über die `AmazonRDSPerformanceInsightsReadOnly`-Richtlinien verfügen, können Sie den Zugriff auf Performance Insights gewähren, indem Sie eine benutzerverwaltete IAM-Richtlinie erstellen oder ändern. Wenn Sie die Richtlinie an einen Berechtigungssatz oder eine Rolle anhängen, kann der Empfänger Performance Insights verwenden.

Erstellen eine benutzerdefinierten Richtlinie

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie auf der Seite Create Policy (Richtlinie erstellen) die Registerkarte „JSON“ aus.
5. Kopieren Sie den folgenden Text und fügen Sie ihn ein. *`us-east-1`* Ersetzen Sie ihn durch den Namen Ihrer AWS Region und *`111122223333`* durch Ihre Kundenkontonummer.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": "rds:DescribeDBInstances",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "rds:DescribeDBClusters",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:DescribeDimensionKeys",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetDimensionKeyDetails",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetResourceMetadata",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:GetResourceMetrics",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:ListAvailableResourceDimensions",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        },
        {
            "Effect": "Allow",
            "Action": "pi:ListAvailableResourceMetrics",
            "Resource": "arn:aws:pi:us-east-1:111122223333:metrics/rds/*"
        }
    ]
}
```

6. Wählen Sie Richtlinie prüfen.

7. Geben Sie einen Namen und optional eine Beschreibung für die Richtlinie an und wählen Sie dann Create policy (Richtlinie erstellen) aus.

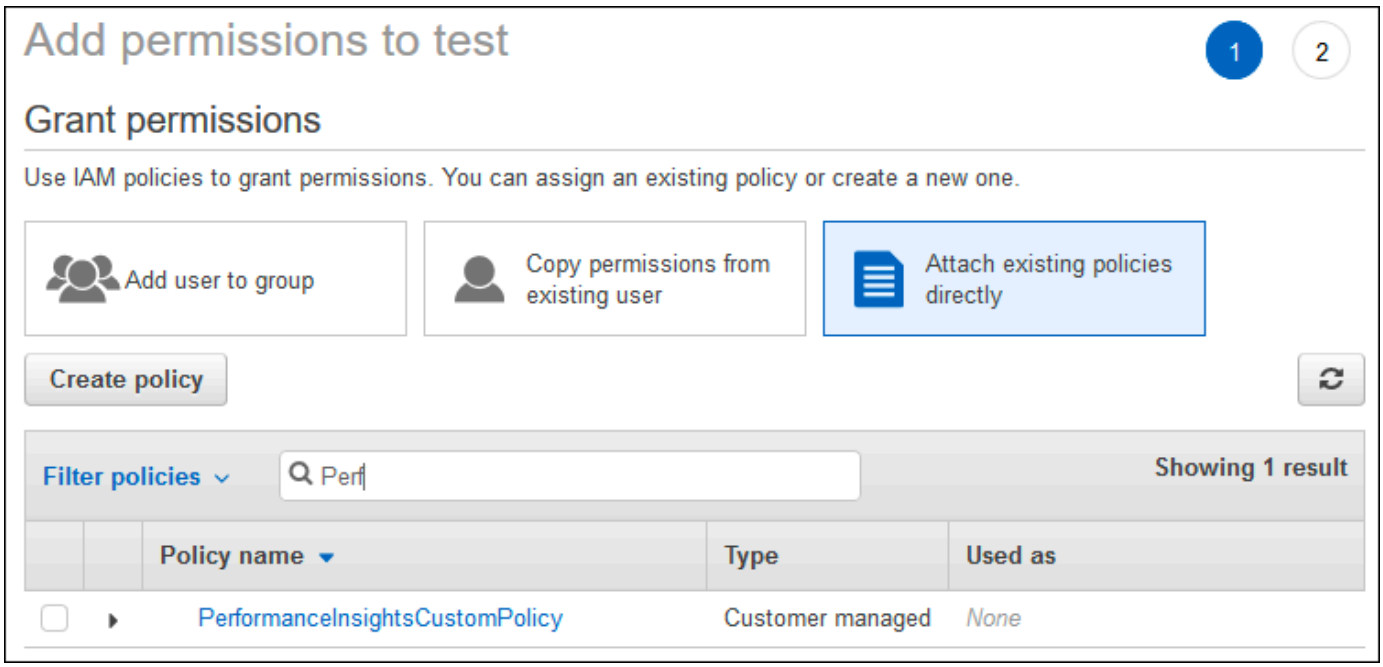
Sie können die Richtlinie nun an einen Berechtigungssatz oder eine Rolle anfügen. Das folgende Verfahren setzt voraus, dass Sie für diesen Zweck bereits einen Benutzer zur Verfügung haben.

So fügen Sie die Richtlinie an einen Benutzer an

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Klicken Sie im Navigationsbereich auf Users (Benutzer).
3. Wählen Sie einen vorhandenen Benutzer aus der Liste aus.

> **Important**
>
> Um Performance Insights verwenden zu können, stellen Sie sicher, dass Sie zusätzlich zur benutzerdefinierten Richtlinie Zugriff auf Amazon DocumentDB haben. Die AmazonDocDBReadOnlyAccessvordefinierte Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf Amazon DoCDB. Weitere Informationen finden Sie unter Zugriff mithilfe von Richtlinien verwalten.

4. Wählen Sie auf der Seite Summary (Übersicht) die Option Add permissions (Berechtigungen hinzufügen) aus.
5. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen). Geben Sie in Suche die ersten Zeichen Ihres Richtliniennamens ein, wie nachfolgend gezeigt.



6. Wählen Sie Ihre Richtlinie und wählen Sie anschließend Nächster Schritt: Prüfen.
7. Wählen Sie Add permissions (Berechtigungen hinzufügen) aus.

Konfiguration einer AWS KMS Richtlinie für Performance Insights

Performance Insights verwendet an AWS KMS key , um sensible Daten zu verschlüsseln. Wenn Sie Performance Insights über die API oder die Konsole aktivieren, haben Sie folgende Möglichkeiten:

- Wählen Sie die Standardeinstellung Von AWS verwalteter Schlüssel.

  Amazon DocumentDB verwendet die Von AWS verwalteter Schlüssel für Ihre neue DB-Instance. Amazon DocumentDB erstellt eine Von AWS verwalteter Schlüssel für Ihr AWS Konto. Ihr AWS Konto hat Von AWS verwalteter Schlüssel für Amazon DocumentDB für jede AWS Region ein anderes.
- Wählen Sie einen kundenverwalteten Schlüssel.

  Wenn Sie einen vom Kunden verwalteten Schlüssel angeben, benötigen Benutzer in Ihrem Konto, die die Performance Insights API aufrufen, die Berechtigungen `kms:Decrypt` und `kms:GenerateDataKey` für den KMS-Schlüssel. Sie können diese Berechtigungen über IAM-Richtlinien konfigurieren. Wir empfehlen jedoch, dass Sie diese Berechtigungen über Ihre KMS-Schlüsselrichtlinie verwalten. Weitere Informationen finden Sie unter Verwenden von Schlüsselrichtlinien in AWS -KMS.

Example

Das folgende Beispiel für eine Schlüsselrichtlinie zeigt, wie Sie Ihrer KMS-Schlüsselrichtlinie Anweisungen hinzufügen können. Diese Anweisungen erlaubt den Zugriff auf Performance Insights. Je nachdem, wie Sie das verwenden AWS KMS, möchten Sie möglicherweise einige Einschränkungen ändern. Bevor Sie Ihrer Richtlinie Anweisungen hinzufügen, entfernen Sie alle Kommentare.

```
{
 "Version" : "2012-10-17",
 "Id" : "your-policy",
 "Statement" : [ {
    //This represents a statement that currently exists in your policy.
 }
 ....,
 //Starting here, add new statement to your policy for Performance Insights.
 //We recommend that you add one new statement for every RDS/DocumentDB instance
 {
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            //One or more principals allowed to access Performance Insights
            "arn:aws:iam::444455556666:role/Role1"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" :{
        "StringEquals" : {
            //Restrict access to only RDS APIs (including Performance Insights).
            //Replace *region* with your AWS Region.
            //For example, specify us-west-2.
            "kms:ViaService" : "rds.*region*.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
            //Restrict access to only data encrypted by Performance Insights.
            "kms:EncryptionContext:aws:pi:service": "rds",
            "kms:EncryptionContext:service": "pi",
```

```
            //Restrict access to a specific DocDB instance.
            //The value is a DbiResourceId.
           "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
        }
    }
}
```

Analyse der Metriken mit dem Performance Insights-Dashboard

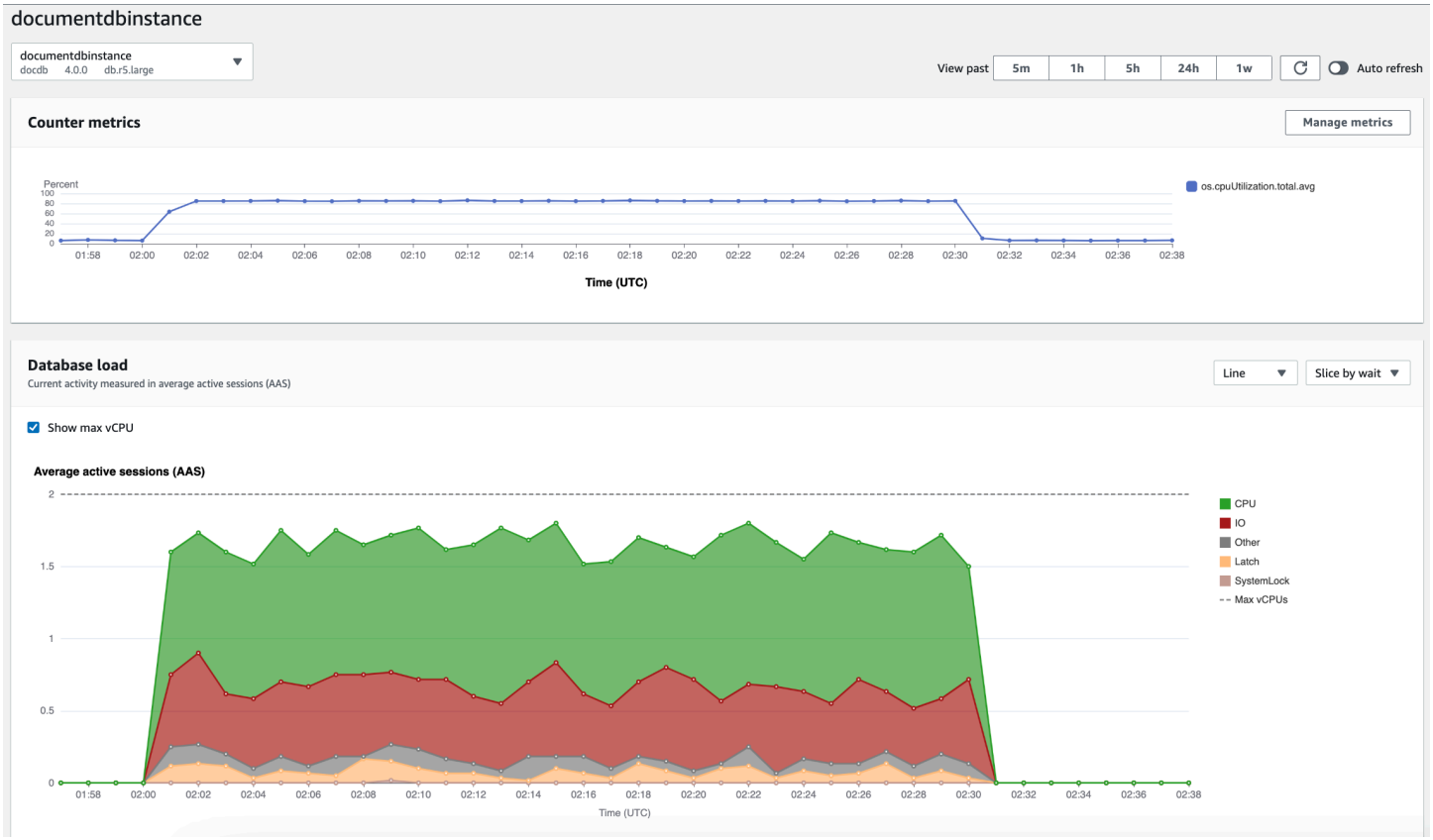
Das Dashboard von Performance Insights enthält Informationen zur Datenbank-Performance, die Sie bei der Analyse und Behebung von Performance-Problemen unterstützen. Auf der Hauptseite des Dashboards können Sie Informationen zur Datenbanklast (DB-Load) einsehen. Sie können die Datenbanklast nach Dimensionen wie Wartestatus oder Abfrage „aufteilen“.

Themen

- Überblick über Performance Insights
- Öffnen des Performance Insights-Dashboards
- Analysieren der Datenbanklast nach Wartezuständen
- Überblick über die Registerkarte „Häufigste Abfragen“
- Das Diagramm zum Laden der Datenbank vergrößern

Überblick über Performance Insights

Das Dashboard ist die einfachste Möglichkeit, mit Performance Insights zu interagieren. Das folgende Beispiel zeigt das Dashboard für eine Amazon DocumentDB DocumentDB-Instance. Standardmäßig zeigt das Dashboard von Performance Insights die Daten der letzten Stunde an.



Das Dashboard ist in folgende Teile gegliedert:

1. Zählermetriken — Zeigt Daten für bestimmte Leistungsindikatormetriken an.
2. Datenbanklast — Zeigt, wie die DB-Last im Vergleich zur DB-Instance-Kapazität abschneidet, wie sie in der Zeile Max vCPU dargestellt wird.
3. Top-Dimensionen — Zeigt die wichtigsten Dimensionen an, die zur DB-Auslastung beitragen. Zu diesen Dimensionen gehören `waits`,`queries`,`hosts`,`databases`, und`applications`.

Themen

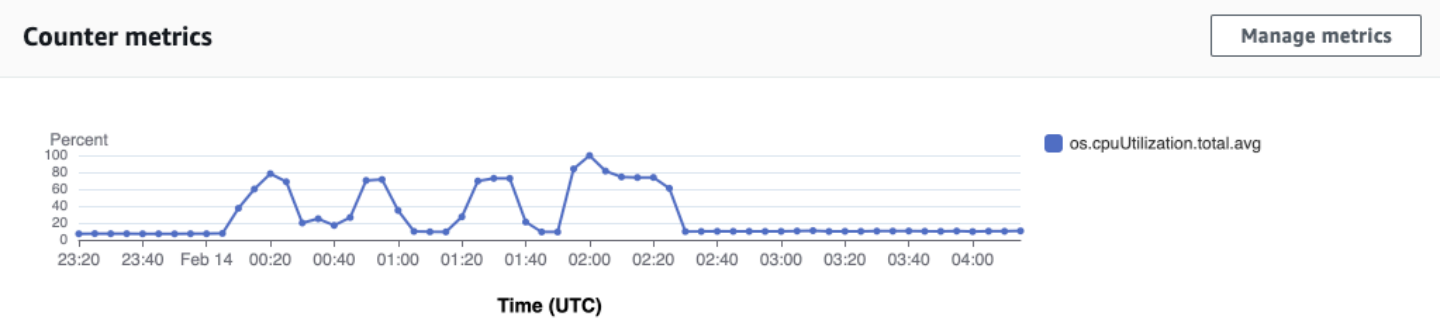
- Zählermetriken-Diagramm
- Datenbank-Ladediagramm
- Dimensionen pro Tabelle

Zählermetriken-Diagramm

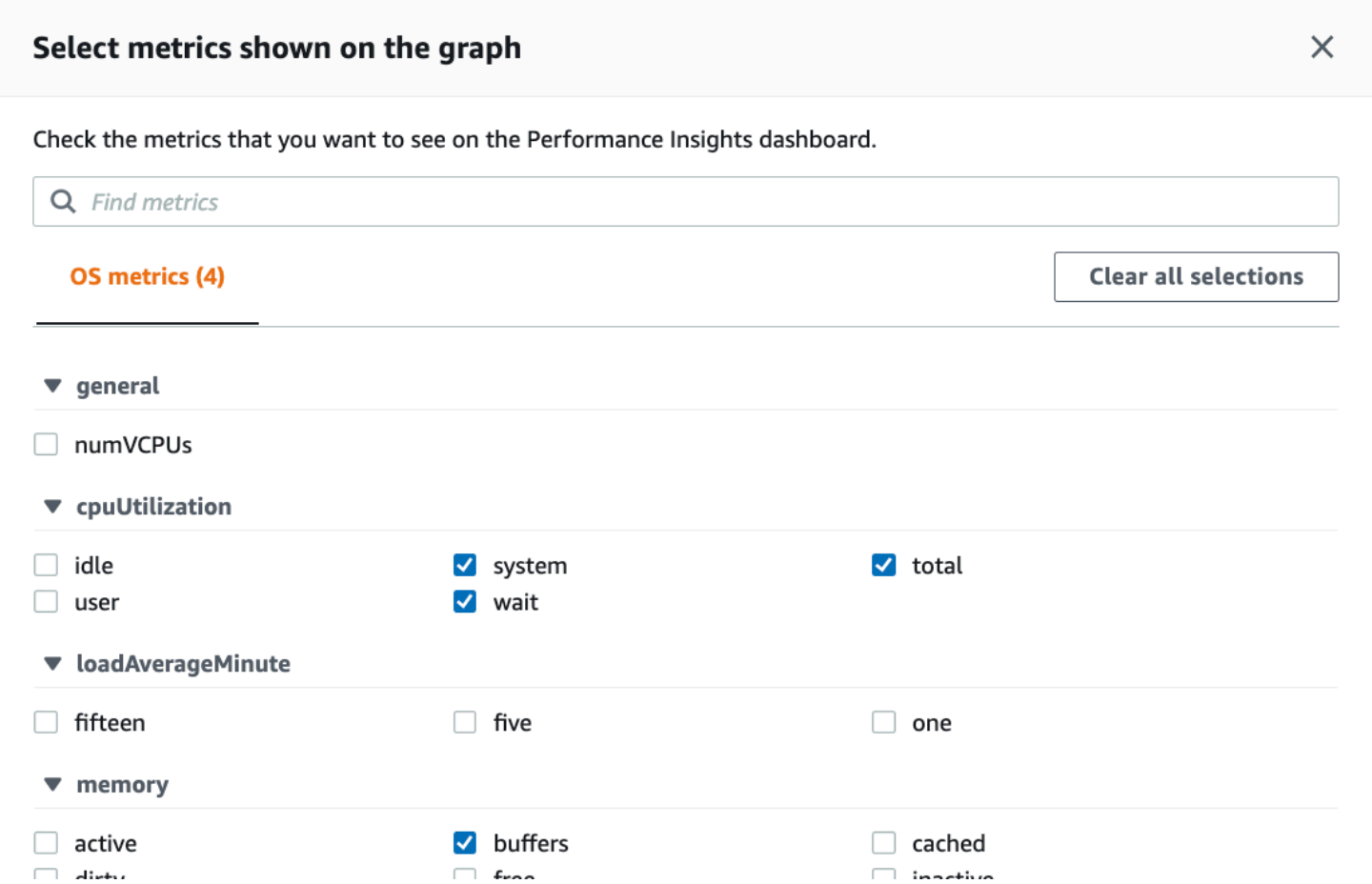
Mithilfe von Zählermetriken können Sie das Performance Insights-Dashboard anpassen und bis zu 10 weitere Diagramme aufnehmen. Diese Grafiken zeigen eine Auswahl von Dutzenden von

Betriebssystemmetriken. Diese Informationen können mit der Datenbanklast korreliert werden, um Performance-Probleme zu identifizieren und zu analysieren.

Das Counter Metrics (Zählermetriken)-Diagramm enthält Daten zu Leistungsindikatoren.



Um die Leistungsindikatoren zu ändern, wählen Sie Metriken verwalten aus. Sie können mehrere Betriebssystemmetriken auswählen, wie im folgenden Screenshot gezeigt. Um Details für jede Metrik anzuzeigen, bewegen Sie den Mauszeiger über den Metriknamen.



Datenbank-Ladediagramm

Das Diagramm zum Laden der Datenbank zeigt, wie die Datenbankaktivität im Vergleich zur Instanzkapazität abschneidet, wie sie in der Zeile Max. vCPU dargestellt wird. Standardmäßig stellt das gestapelte Liniendiagramm die DB-Last als durchschnittliche aktive Sitzungen pro Zeiteinheit dar. Die DB-Last wird nach Wartestatus aufgeteilt (gruppiert).



DB-Last aufgeteilt nach Dimensionen

Sie können die Last als aktive Sitzungen anzeigen, die nach unterstützten Dimensionen gruppiert sind. Die folgende Abbildung zeigt die Abmessungen für die Amazon DocumentDB DocumentDB-Instance.

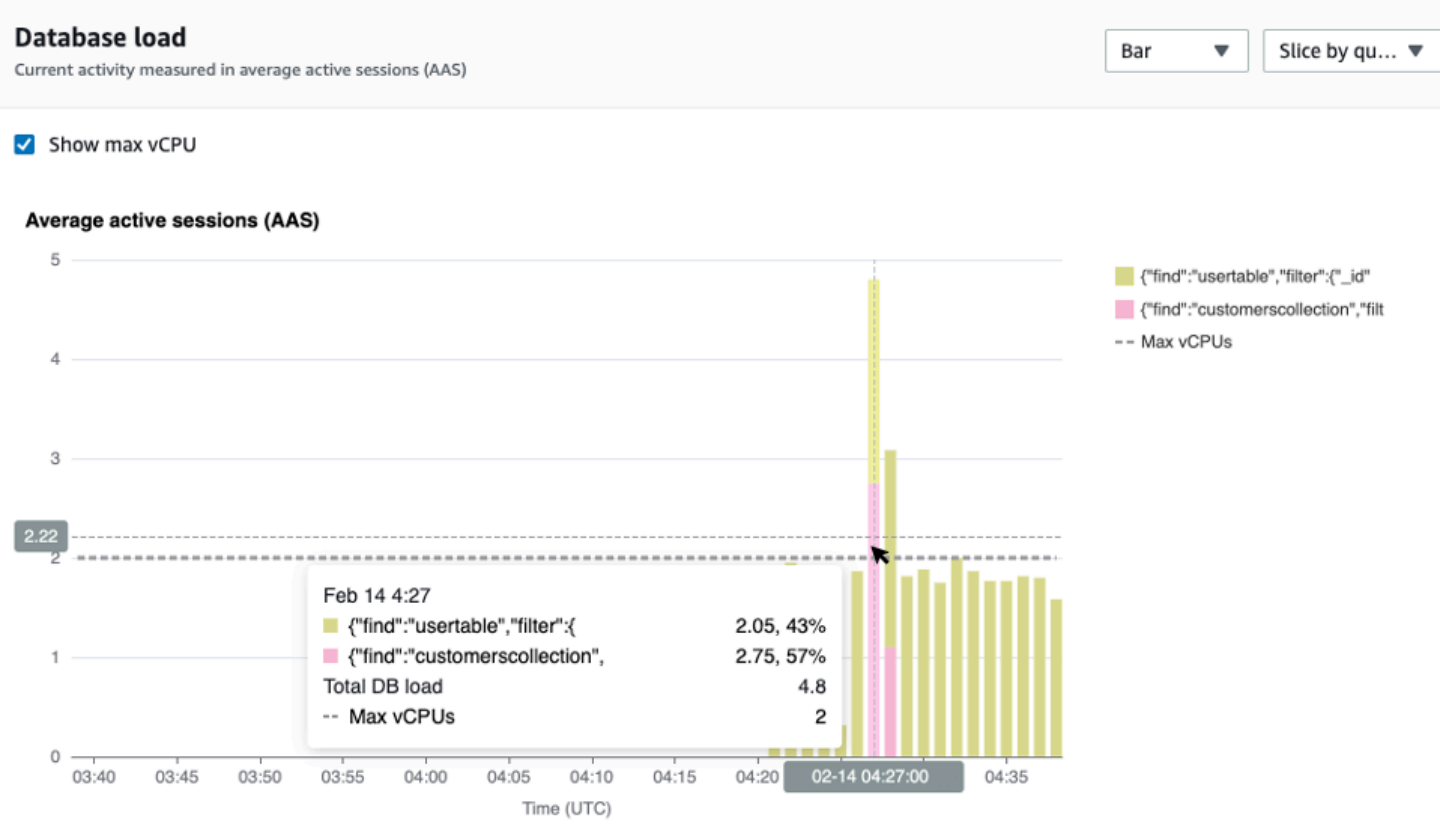


DB-Ladedetails für ein Dimensionselement

Um Details zu einem DB-Lastelement innerhalb einer Dimension anzuzeigen, bewegen Sie den Mauszeiger über den Elementnamen. Die folgende Abbildung zeigt Details für eine Abfrageanweisung.



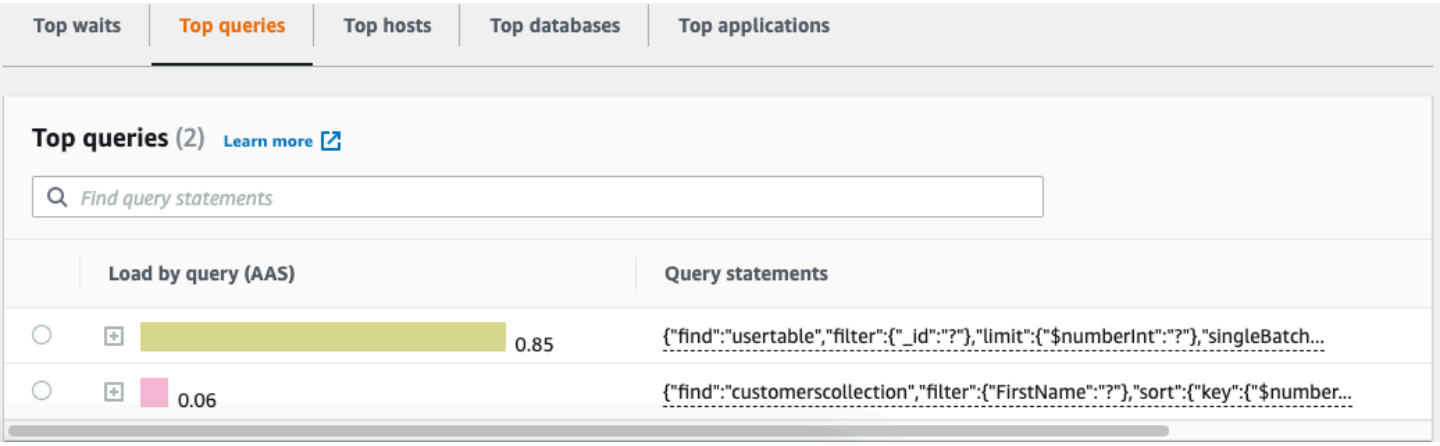
Um Details zu einem Element für den ausgewählten Zeitraum in der Legende anzuzeigen, bewegen Sie den Mauszeiger über dieses Element.



Dimensionen pro Tabelle

In der Tabelle mit den obersten Abmessungen wird die Datenbanklast nach verschiedenen Dimensionen aufgeteilt. Eine Dimension ist eine Kategorie oder „Aufteilung“ für verschiedene Merkmale der DB-Last. Wenn es sich bei der Dimension um eine Abfrage handelt, werden in den häufigsten Abfragen die Abfrageanweisungen angezeigt, die am meisten zur DB-Auslastung beitragen.

Wählen Sie eine der folgenden Dimensionsregisterkarten.



Die folgende Tabelle enthält eine kurze Beschreibung der einzelnen Registerkarten.

Tab	Beschreibung
Das Top-Ereignis (auf der Datenbank-Backend wartet)	Das Top-Ereignis, auf das die Datenbank wartet
Die Top-Abfragen	Die Abfragen, Anweisungen oder Abfragen, die derzeit ausgeführt werden
Die Top-Hosts (IP-Adressen oder Host-Ports)	Die Top-Hosts, die mit dem Port des Clients verbunden sind
Die Top-Datenbanken	Die Top-Datenbanken

Beschreibung
(Top-Datenbanken) mit der der Client verbunden ist
(Top-Anwendungen) mit der Datenbank verbunden ist

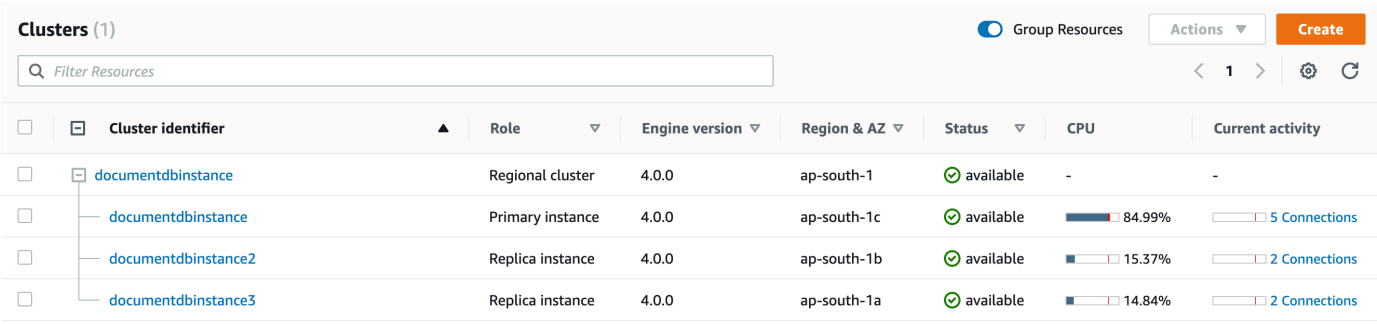
Informationen zum Analysieren von Abfragen mithilfe der Registerkarte „Häufig gestellte Abfragen“ finden Sie unter Überblick über die Registerkarte „Häufigste Abfragen“.

Öffnen des Performance Insights-Dashboards

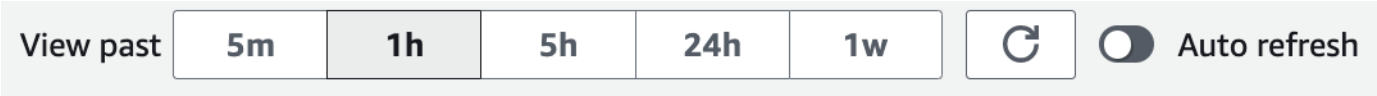
Gehen Sie wie folgt vor, um das Performance Insights Insights-Dashboard in der AWS Management Console anzuzeigen:

1. Öffnen Sie die Performance Insights Insights-Konsole unter https://console.aws.amazon.com/docdb/.
2. Wählen Sie eine DB-Instance aus. Das Performance Insights Insights-Dashboard wird für diese Amazon DocumentDB DocumentDB-Instance angezeigt.

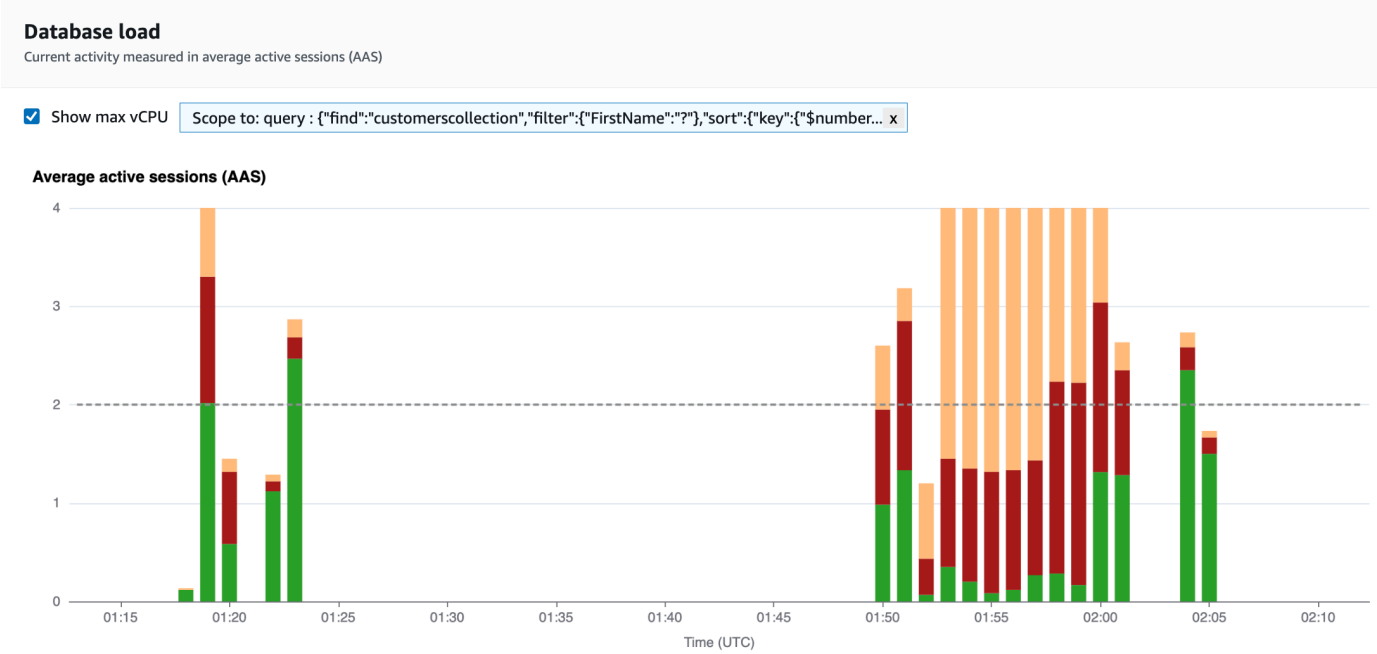
Für Amazon DocumentDB DocumentDB-Instances mit aktiviertem Performance Insights können Sie das Dashboard auch aufrufen, indem Sie in der Liste der Instances den Eintrag Sessions auswählen. Unter Aktuelle Aktivität zeigt das Element Sitzungen die Datenbanklast von durchschnittlichen, aktiven Sitzungen der letzten fünf Minuten an. Der Balken zeigt die Last grafisch an. Wenn die Leiste leer ist, befindet sich die Instance im Leerlauf. Wenn die Last ansteigt, wird der Balken blau ausgefüllt. Wenn die Last die Anzahl virtueller CPUs (vCPUs) in der Instanzklasse überschreitet, färbt sich der Balken rot, was auf einen potenziellen Engpass hinweist.



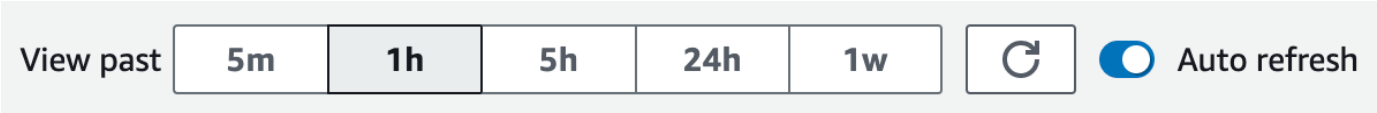
3. (Optional) Wählen Sie über eine der Schaltflächen rechts oben ein anderes Zeitintervall aus. Um das Intervall beispielsweise auf 1 Stunde zu ändern, wählen Sie 1h aus.



Im folgenden Screenshot beträgt das DB-Ladeintervall 1 Stunde.



4. Um Ihre Daten automatisch zu aktualisieren, aktivieren Sie Automatische Aktualisierung.



   Das Performance Insights-Dashboard wird automatisch mit neuen Daten aktualisiert. Die Aktualisierungsrate hängt von der Menge der angezeigten Daten ab:

   - „5 Minuten“ wird alle 5 Sekunden aktualisiert.
   - 1 Stunde, wird jede Minute aktualisiert.
   - 5 Stunden, wird jede Minute aktualisiert.
   - „24 Stunden“ wird alle 5 Minuten aktualisiert.
   - „1 Woche“ wird jede Stunden aktualisiert.

Analysieren der Datenbanklast nach Wartezuständen

Wenn das Diagramm zur Datenbankauslastung (DB-Last) einen Engpass anzeigt, können Sie herausfinden, woher die Last kommt. Betrachten Sie dazu die Tabelle mit den Hauptlastelementen unterhalb des Datenbanklast-Diagramms Wählen Sie ein bestimmtes Element aus, z. B. eine Abfrage oder eine Anwendung, um dieses Element genauer zu untersuchen und Details zu diesem Element anzuzeigen.

Die Datenbanklast, gruppiert nach Waits und Top-Abfragen, bietet in der Regel den besten Einblick in Leistungsprobleme. DB-Last gruppiert nach Wartezeiten zeigt an, ob Ressourcen- oder Parallelitätsengpässe in der Datenbank vorhanden sind. In diesem Fall wird auf der Registerkarte „Häufigste Abfragen“ der Tabelle mit den am häufigsten ausgelasteten Elementen angezeigt, welche Abfragen für diese Auslastung verantwortlich sind.

Ihr typischer Workflow für die Diagnose von Performance-Problemen ist folgendermaßen:

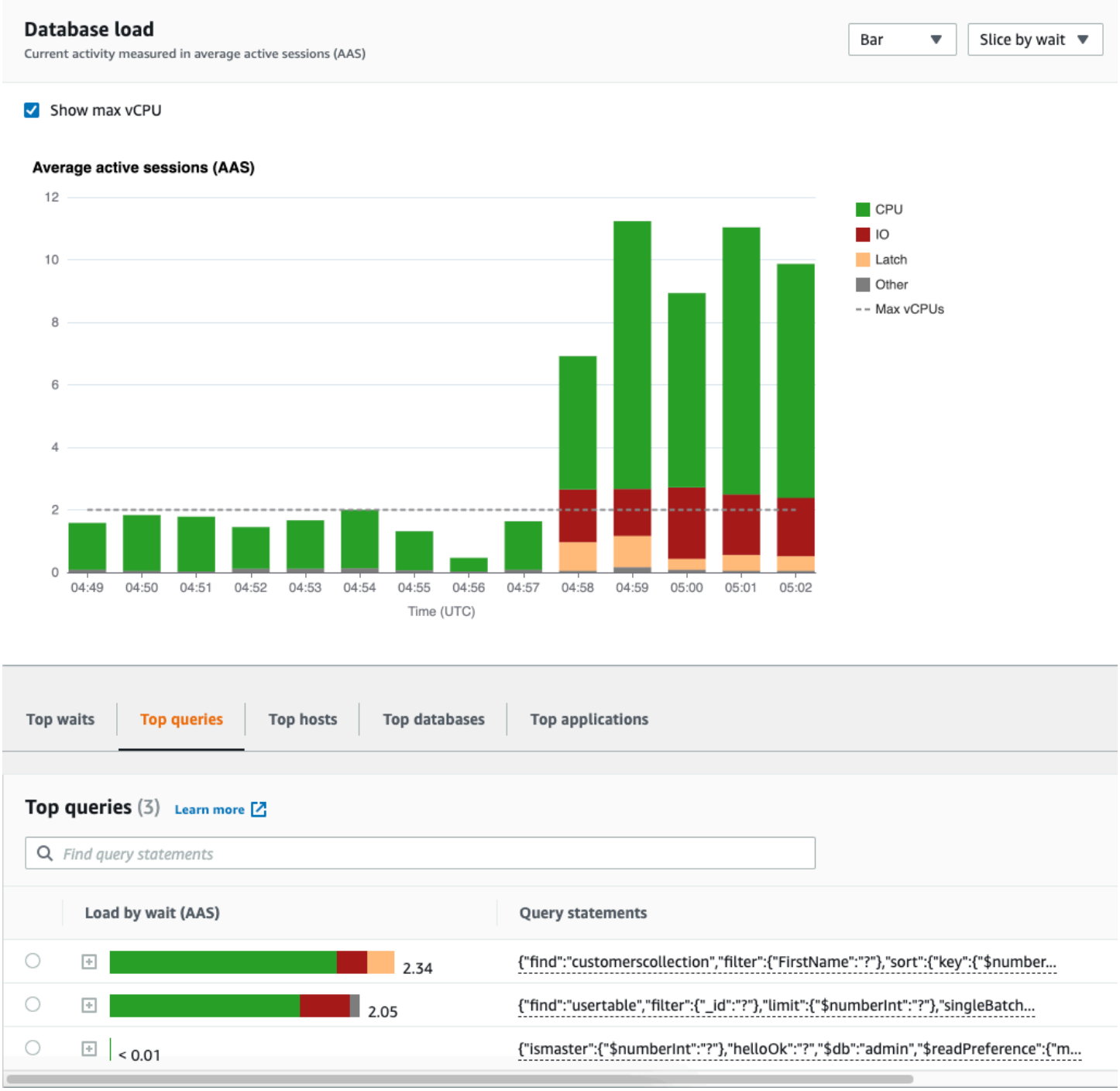
1. Überprüfen Sie das Diagramm der durchschnittlich aktiven Sitzungen auf irgendwelche Ereignisse, in denen die Datenbanklast die Max CPU-Linie übersteigt.
2. Wenn ja, schauen Sie sich das Diagramm der durchschnittlich aktiven Sitzungen an und identifizieren Sie, welcher Wartezustand oder welche Zustände primär dafür verantwortlich sind.
3. Identifizieren Sie die Digest-Abfragen, die die Last verursacht haben, indem Sie sehen, welche der Abfragen auf der Registerkarte „Häufigste Abfragen“ in der Tabelle mit den am häufigsten

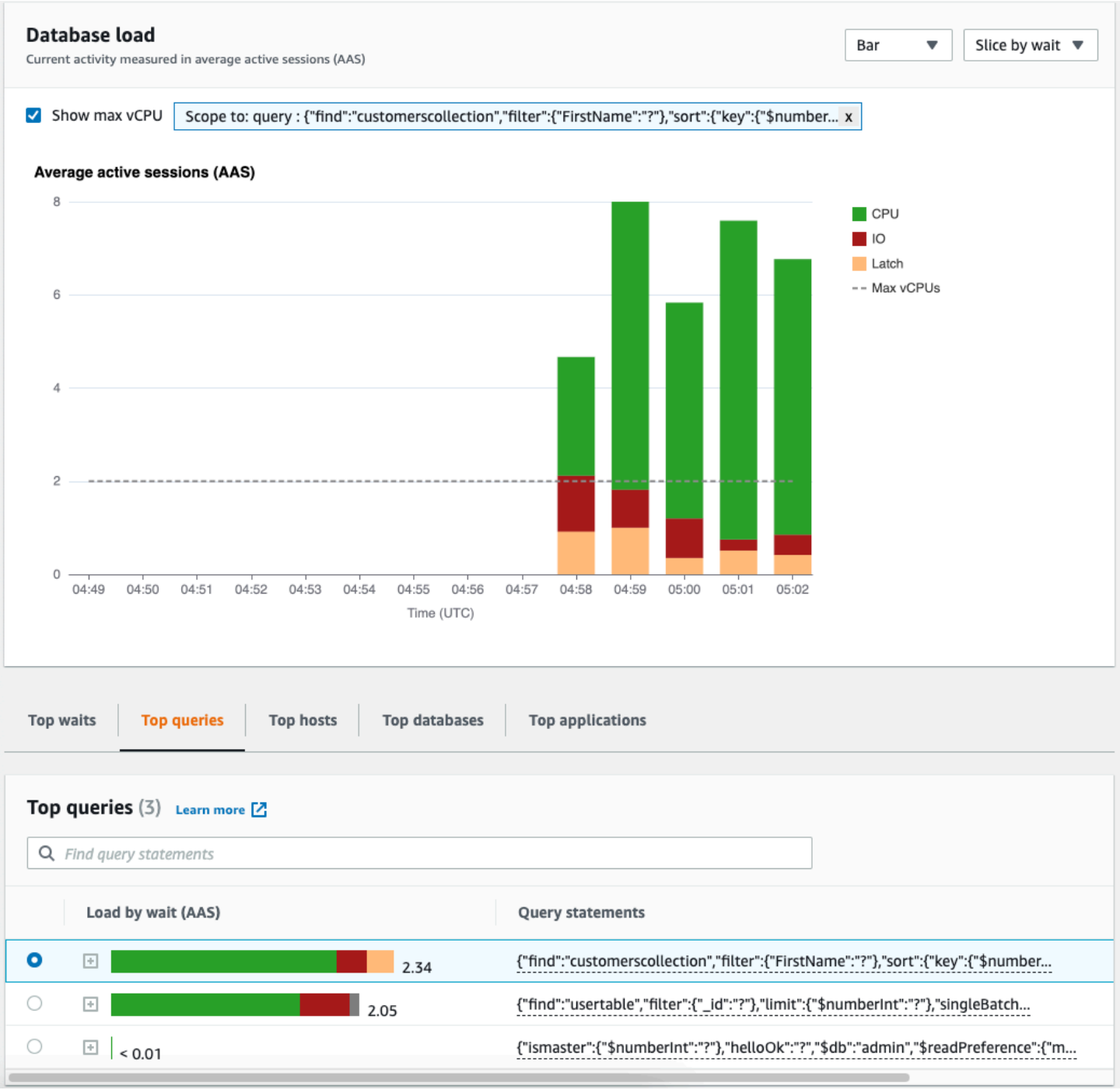
geladenen Elementen am meisten zu diesen Wartezuständen beitragen. Sie können diese anhand der Spalte Load by Wait (AAS) identifizieren.

4. Wählen Sie auf der Registerkarte „Häufig gestellte Abfragen“ eine dieser Digest-Abfragen aus, um sie zu erweitern und die untergeordneten Abfragen zu sehen, aus denen sie besteht.

Sie können auch sehen, welche Hosts oder Anwendungen die meiste Last verursachen, indem Sie jeweils Top-Hosts oder Top-Anwendungen auswählen. Anwendungsnamen werden in der Verbindungszeichenfolge zur Amazon DocumentDB DocumentDB-Instance angegeben. `Unknown`gibt an, dass das Anwendungsfeld nicht angegeben wurde.

Im folgenden Dashboard machen beispielsweise CPU-Wartezeiten den größten Teil der DB-Last aus. Wenn Sie unter Häufigste Abfragen die oberste Abfrage auswählen, wird das Diagramm zur Datenbankauslastung so ausgerichtet, dass der Schwerpunkt auf der höchsten Last liegt, die durch die ausgewählte Abfrage verursacht wird.


Database load
Current activity measured in average active sessions (AAS)
Bar
Slice by wait
Show max vCPU
Average active sessions (AAS)
12
10
8
6
4
2
0
CPU
IO
Latch
Other
Max vCPUs
04:49
04:50
04:51
04:52
04:53
04:54
04:55
04:56
04:57
04:58
04:59
05:00
05:01
05:02
Time (UTC)
Top waits
Top queries
Top hosts
Top databases
Top applications
Top queries (3)
Learn more
Find query statements
Load by wait (AAS)
Query statements
2.34
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":{"$number...
2.05
{"find":"usertable","filter":{"_id":"?"},"limit":{"$numberInt":"?"},"singleBatch...
< 0.01
{"ismaster":{"$numberInt":"?"},"helloOk":"?","$db":"admin","$readPreference":{"m...



Überblick über die Registerkarte „Häufigste Abfragen“

Standardmäßig werden auf der Registerkarte „Häufigste Abfragen“ die Abfragen angezeigt, die am meisten zur DB-Auslastung beitragen. Sie können den Abfragetext analysieren, um Ihre Abfragen zu optimieren.

Themen

- Zusammenfassungen von Abfragen
- Nach Waits laden (AAS)
- Detaillierte Abfrageinformationen anzeigen
- Zugriff auf den Abfragetext der Anweisung
- Abfragetext für Kontoauszüge anzeigen und herunterladen

Zusammenfassungen von Abfragen

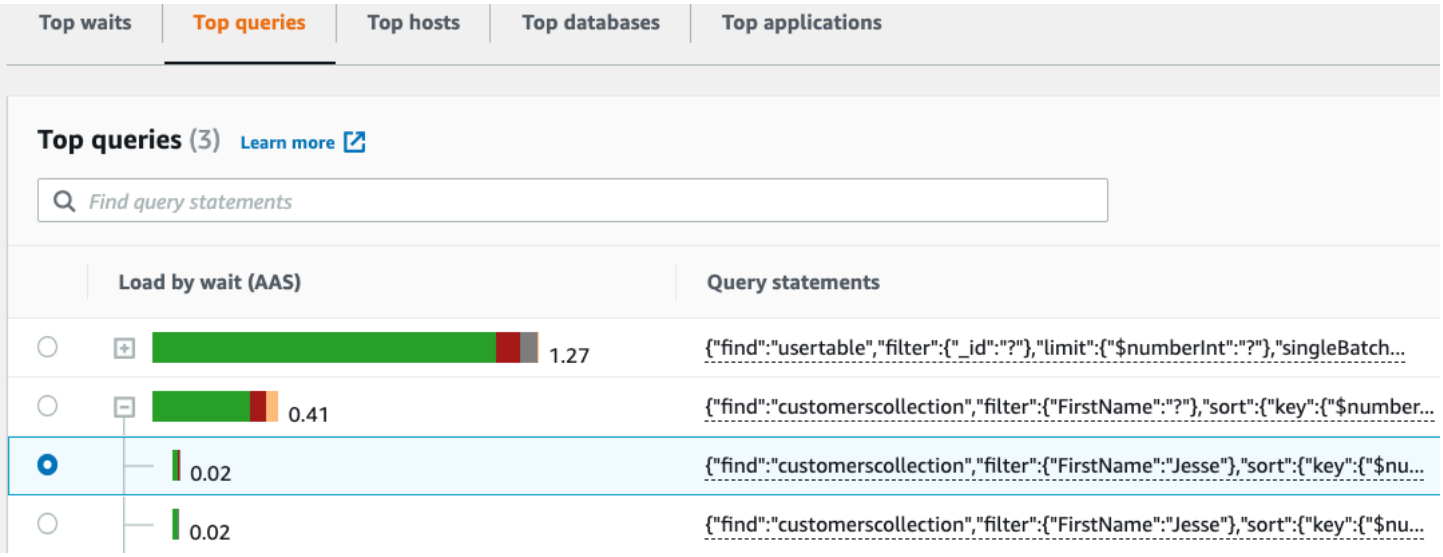
Ein Abfrage-Digest besteht aus mehreren tatsächlichen Abfragen, die sich strukturell ähneln, aber unterschiedliche Literalwerte haben können. Der Digest ersetzt fest codierte Werte durch ein Fragezeichen. Ein Abfrage-Digest könnte beispielsweise so aussehen:

```
{"find":"customerscollection","filter":{"FirstName":"?"},"sort":{"key":
{"$numberInt":"?"}},"limit":{"$numberInt":"?"}}
```

Dieser Digest kann die folgenden untergeordneten Abfragen enthalten:

```
{"find":"customerscollection","filter":{"FirstName":"Karrie"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Met"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
{"find":"customerscollection","filter":{"FirstName":"Rashin"},"sort":{"key":
{"$numberInt":"1"}},"limit":{"$numberInt":"3"}}
```

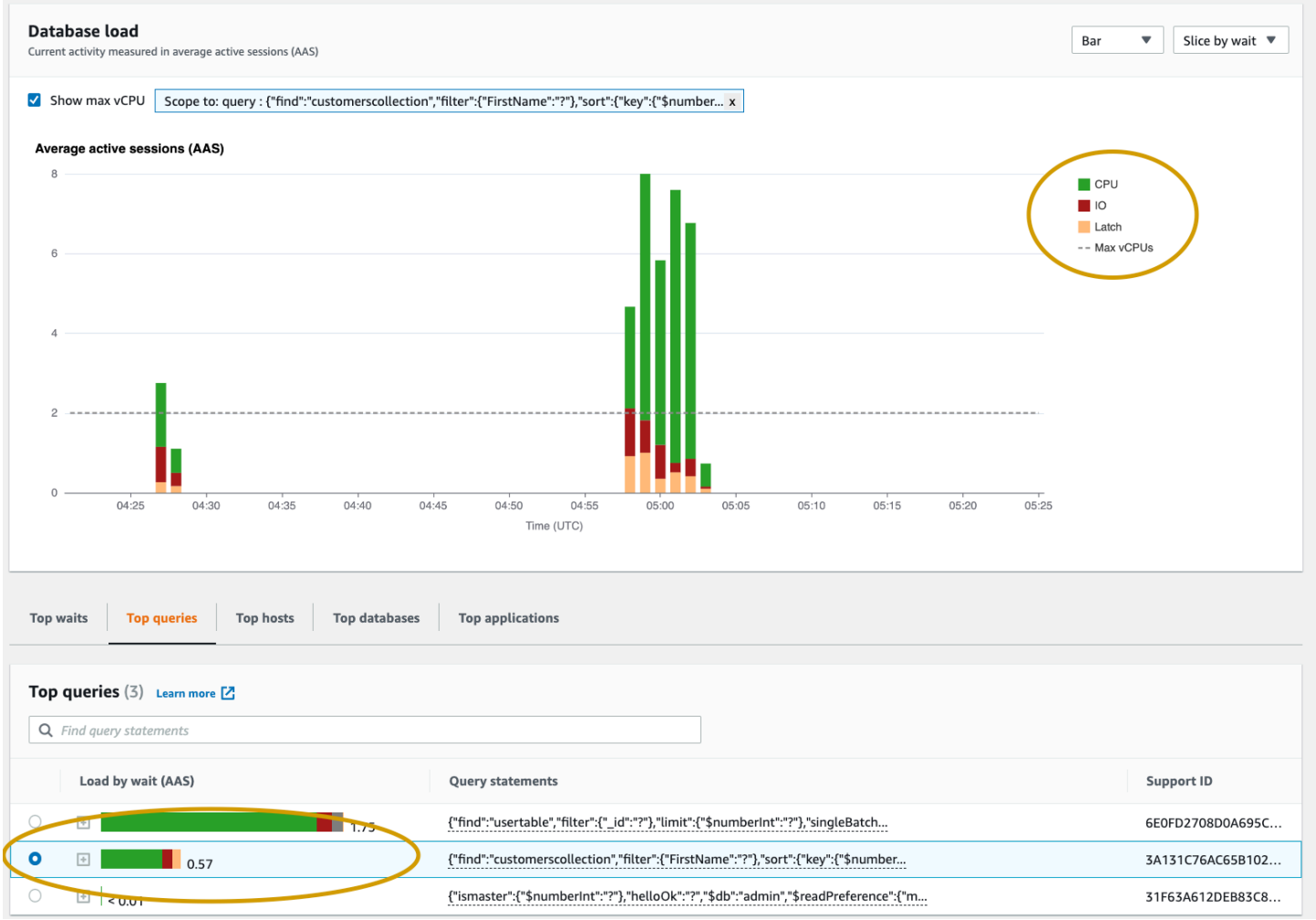
Um die wörtlichen Abfrageanweisungen in einem Digest zu sehen, wählen Sie die Abfrage und dann das Pluszeichen () aus. + Im folgenden Screenshot ist die ausgewählte Abfrage ein Digest.



> **Note**
>
> In einem Abfrage-Digest werden ähnliche Abfrageanweisungen gruppiert, sensible Informationen werden jedoch nicht geschwärzt.

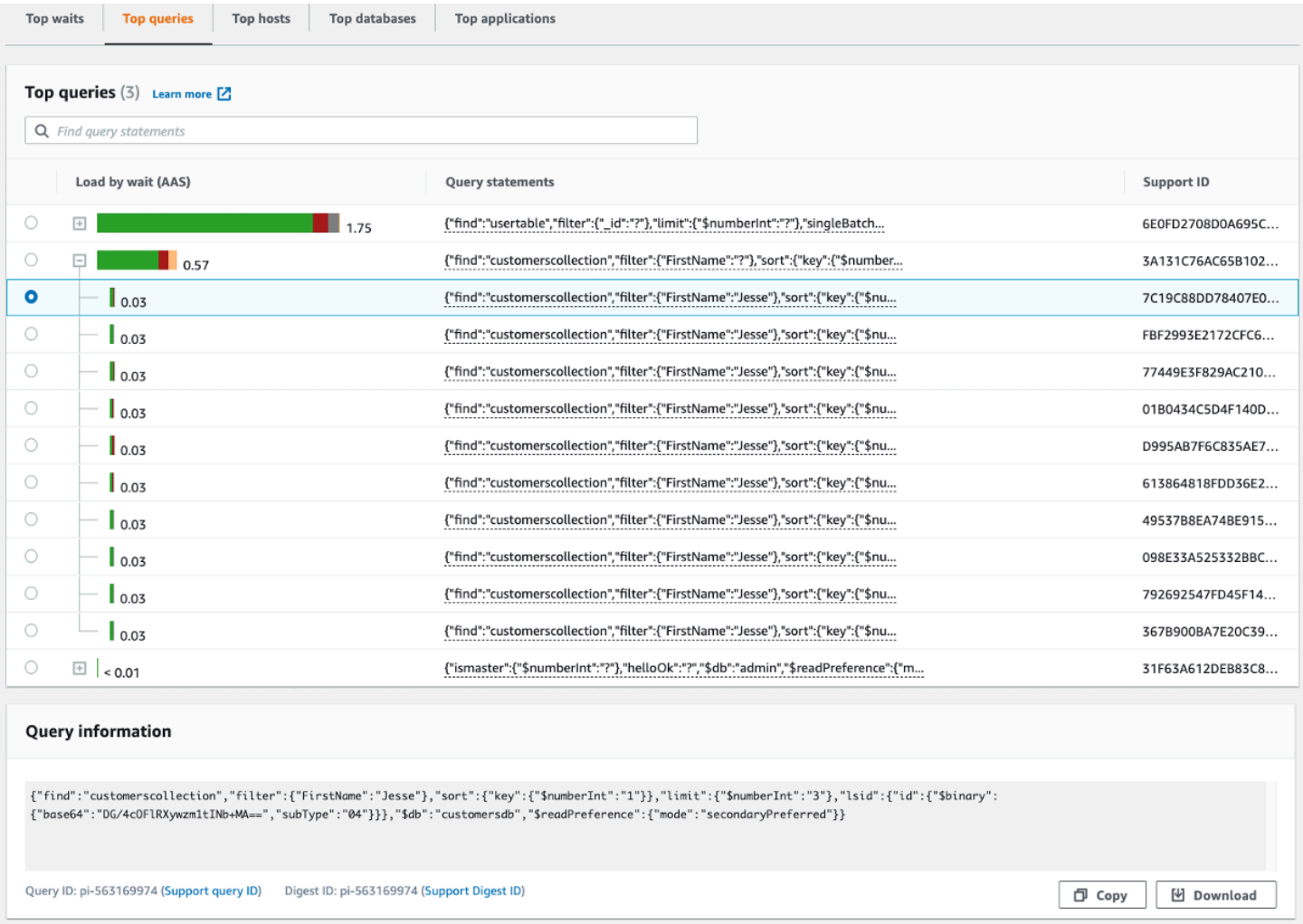
Nach Waits laden (AAS)

In Top-Abfragen zeigt die Spalte Load by Waits (AAS) den Prozentsatz der Datenbanklast, der jedem Top-Load-Element zugeordnet ist. Diese Spalte gibt die Last für dieses Element nach der Gruppierung wieder, die derzeit im DB-Lastdiagramm ausgewählt ist. Beispielsweise können Sie das DB-Last-Diagramm nach Wartezuständen gruppieren. In diesem Fall ist der Balken DB Load by Waits (DB-Last nach Wartezuständen) so groß, segmentiert und farbcodiert, dass angezeigt wird, zu wieviel Prozent diese Abfrage zum betreffenden Wartezustand beiträgt. Es zeigt zudem auf, welche Wartezustände sich auf die ausgewählte Abfrage auswirken.



Detaillierte Abfrageinformationen anzeigen

In der Abfragetabelle „Top“ können Sie eine Digest-Anweisung öffnen, um die zugehörigen Informationen einzusehen. Die Informationen werden im unteren Bereich angezeigt.



Die folgenden Typen von Bezeichnern (IDs) sind mit Abfrageanweisungen verknüpft:

1. Support-Abfrage-ID — Ein Hashwert der Abfrage-ID. Dieser Wert dient nur zum Verweisen auf eine Abfrage-ID, wenn Sie mit AWS Support arbeiten. AWS Der Support hat keinen Zugriff auf Ihre tatsächliche Anfrage IDs und Ihren Abfragetext.
2. Support-Digest-ID — Ein Hashwert der Digest-ID. Dieser Wert dient nur zum Verweisen auf eine Digest-ID, wenn Sie mit AWS Support zusammenarbeiten. AWS Der Support hat keinen Zugriff auf Ihre tatsächliche Zusammenfassung IDs und Ihren Abfragetext.

Zugriff auf den Abfragetext der Anweisung

Standardmäßig enthält jede Zeile in der Tabelle mit den häufigsten Abfragen 500 Byte Abfragetext für jede Abfrageanweisung. Wenn eine Digest-Anweisung 500 Byte überschreitet, können Sie mehr Text anzeigen, indem Sie die Anweisung im Performance Insights Insights-Dashboard öffnen. In

diesem Fall beträgt die maximale Länge der angezeigten Abfrage 1 KB. Wenn Sie eine vollständige Abfrageanweisung anzeigen, können Sie auch Herunterladen wählen.

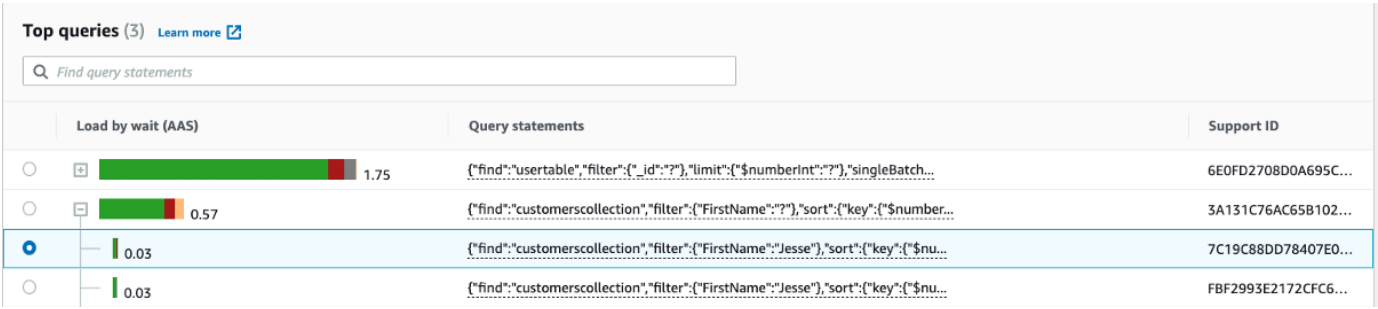
Abfragetext für Kontoauszüge anzeigen und herunterladen

Im Performance Insights Insights-Dashboard können Sie Abfragetext anzeigen oder herunterladen.

Um mehr Abfragetext im Performance Insights Insights-Dashboard anzuzeigen

1. Öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter: https://console.aws.amazon.com/docdb/
2. Wählen Sie im Navigationsbereich Performance-Insights aus.
3. Wählen Sie eine DB-Instance aus. Das Performance Insights-Dashboard wird für diese DB-Instance angezeigt.

   Abfrageanweisungen mit Text, der größer als 500 Byte ist, sehen wie in der folgenden Abbildung aus:



4. Sehen Sie sich den Abschnitt mit den Abfrageinformationen an, um einen größeren Teil des Abfragetextes zu sehen.



Das Performance Insights Insights-Dashboard kann bis zu 1 KB für jede vollständige Abfrageanweisung anzeigen.

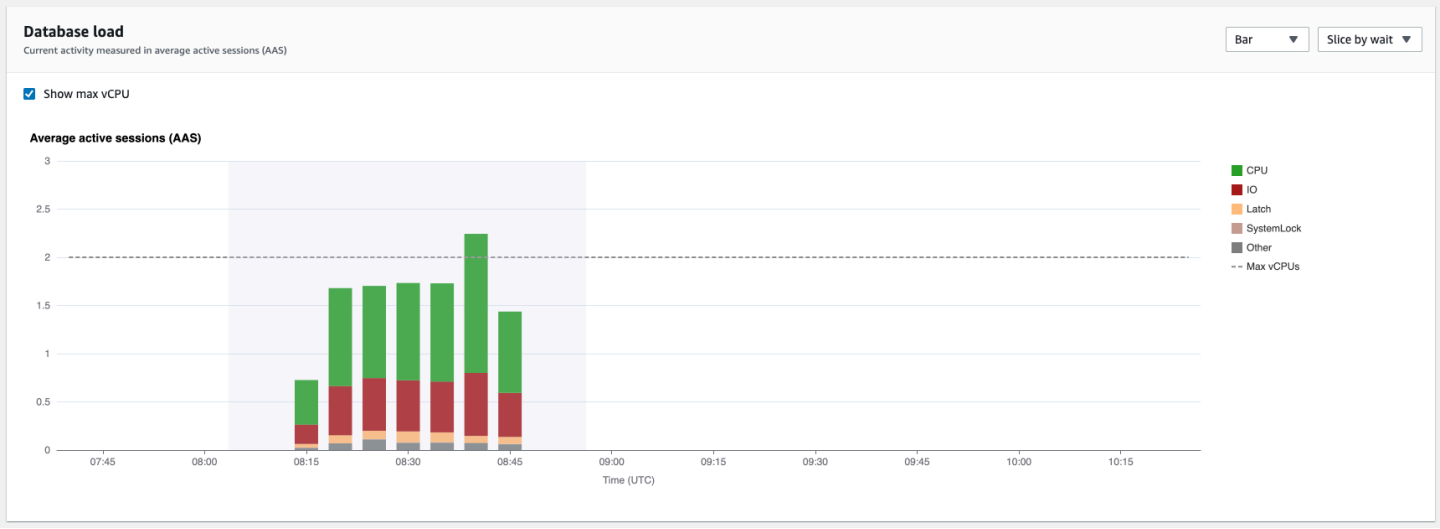
> **Note**
>
> Um die Abfrageanweisung zu kopieren oder herunterzuladen, deaktivieren Sie alle Popup-Blocker.

Das Diagramm zum Laden der Datenbank vergrößern

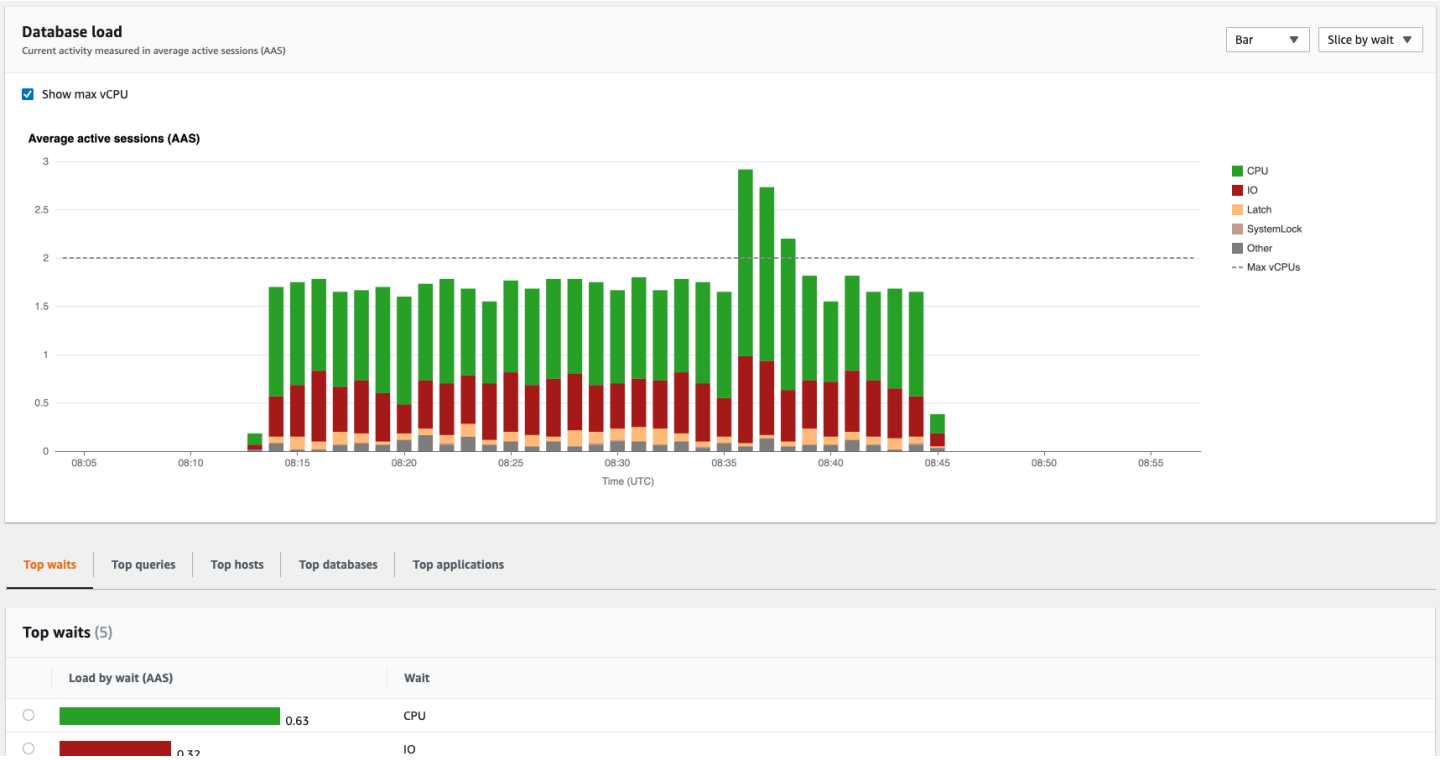
Sie können weitere Funktionen der Benutzeroberfläche von Performance Insights verwenden, um die Performance-Daten zu analysieren.

Click-and-Drag Vergrößern

In der Benutzeroberfläche von Performance Insights können Sie einen kleinen Teil des Lastdiagramms auswählen und die Details vergrößern.



Um einen Teil des Lastdiagramms zu vergrößern, wählen Sie die Startzeit und ziehen Sie mit der Maus an das Ende des gewünschten Zeitraums. Dabei wird der markierte Bereich farblich hervorgehoben. Wenn Sie die Maustaste loslassen, vergrößert das Lastdiagramm den ausgewählten Bereich, und die Tabelle mit den wichtigsten Elementen wird neu berechnet.



Abrufen von Metriken mit der Performance Insights-API

Wenn Performance Insights aktiviert ist, bietet die API Einblicke in die Instance-Leistung Amazon CloudWatch Logs bietet die maßgebliche Quelle für verkaufte Monitoring-Metriken für AWS Services.

Performance Insights bietet eine domänenspezifische Ansicht der Datenbanklast, gemessen als durchschnittliche aktive Sitzungen (AAS). Diese Metrik erscheint API-Verbrauchern als zweidimensionaler Zeitreihendatensatz. Die Zeitdimension der Daten stellt die Datenbanklastdaten für jeden Zeitpunkt im abgefragten Zeitraum bereit. Für jeden Zeitpunkt wird die Gesamtlast bezogen auf die angeforderten Dimensionen zerlegt, z. B. `Query`, `Wait-state`, `Application` oder `Host`, gemessen zum betreffenden Zeitpunkt.

Amazon DocumentDB Performance Insights überwacht Ihre Amazon DocumentDB-DB-Instance, sodass Sie die Datenbankleistung analysieren und Fehler beheben können. Eine Möglichkeit zum Anzeigen von Performance Insights-Daten bietet die AWS Management Console. Performance Insights stellt außerdem eine öffentliche API bereit, sodass Sie Ihre eigenen Daten abfragen können. Sie können die API für Folgendes verwenden:

- Auslagern von Daten in eine Datenbank
- Hinzufügen von Performance Insights-Daten zu bestehenden Überwachungs-Dashboards
- Entwickeln von Überwachungstools

Um die Performance Insights-API zu verwenden, aktivieren Sie Performance Insights auf einer Ihrer Amazon DocumentDB DocumentDB-Instances. Weitere Informationen zum Aktivieren von Performance Insights finden Sie unter Aktivieren und Deaktivieren von Performance Insights. Weitere Informationen zur Performance Insights-API finden Sie in der Referenz zur Performance Insights-API.

Die Performance Insights-API bietet die folgenden Operationen.

Performance-Insights-Aktion	AWS CLI Befehl	Beschreibung
DescribeDimensionKeys	aws pi describe-dimension-keys	Ruft die Schlüssel der Top N-Dimension für eine Metrik für einen bestimmten Zeitraum ab.
GetDimensionKeyDetails	aws pi get-dimension-key-details	Ruft die Attribute der angegebenen Dimensionsgruppe für eine DB-Instance oder Datenquelle ab. Wenn Sie beispielsweise eine Abfrage-ID angeben und die Dimensionsdetails verfügbar sind, wird der vollständige Text der Dimension `GetDimensionKeyDetails` abgerufen, die dieser ID `db.query.statement` zugeordnet ist. Dieser Vorgang ist nützlich, weil er `GetResourceMetrics` das Abrufen von umfangreichem Text in Abfrageanweisungen `DescribeDimensionKeys` nicht unterstützt.
GetResourceMetadata	aws pi get-resource-metadata	Rufen Sie die Metadaten für verschiedene Funktionen ab. Die Metadaten könnten

Performance-Insights-Aktion	AWS CLI Befehl	Beschreibung
		beispielsweise darauf hindeuten, dass eine Funktion für eine bestimmte DB-Instan ce ein- oder ausgeschaltet ist.
GetResourceMetrics	aws pi get-resource-metrics	Ruft Performance Insights-Metriken für eine Reihe von Datenquellen über einen Zeitraum ab. Sie können spezifische Dimension sgruppen und Dimensionen bereitstellen und Aggregati on und Filterkriterien für jede Gruppe bereitstellen.
ListAvailableResou rceDimensions	aws pi list-avai lable-resource-dim ensions	Rufen Sie die Dimensionen ab, die für jeden angegebenen Metriktyp für eine bestimmte Instance abgefragt werden können.
ListAvailableResou rceMetrics	aws pi list-avai lable-resource-met rics	Rufen Sie alle verfügbaren Metriken der angegeben en Metriktypen ab, die für eine bestimmte DB-Instance abgefragt werden können.

Themen

- AWS CLI für Performance Insights
- Abrufen von Zeitreihenmetriken
- AWS CLI Beispiele für Performance Insights

AWS CLI für Performance Insights

Sie können Performance Insights-Daten über die anzeige AWS CLI. Hilfe zu den AWS CLI -Befehlen für Performance Insights erhalten Sie durch Eingabe der folgenden Befehle an der Befehlszeile.

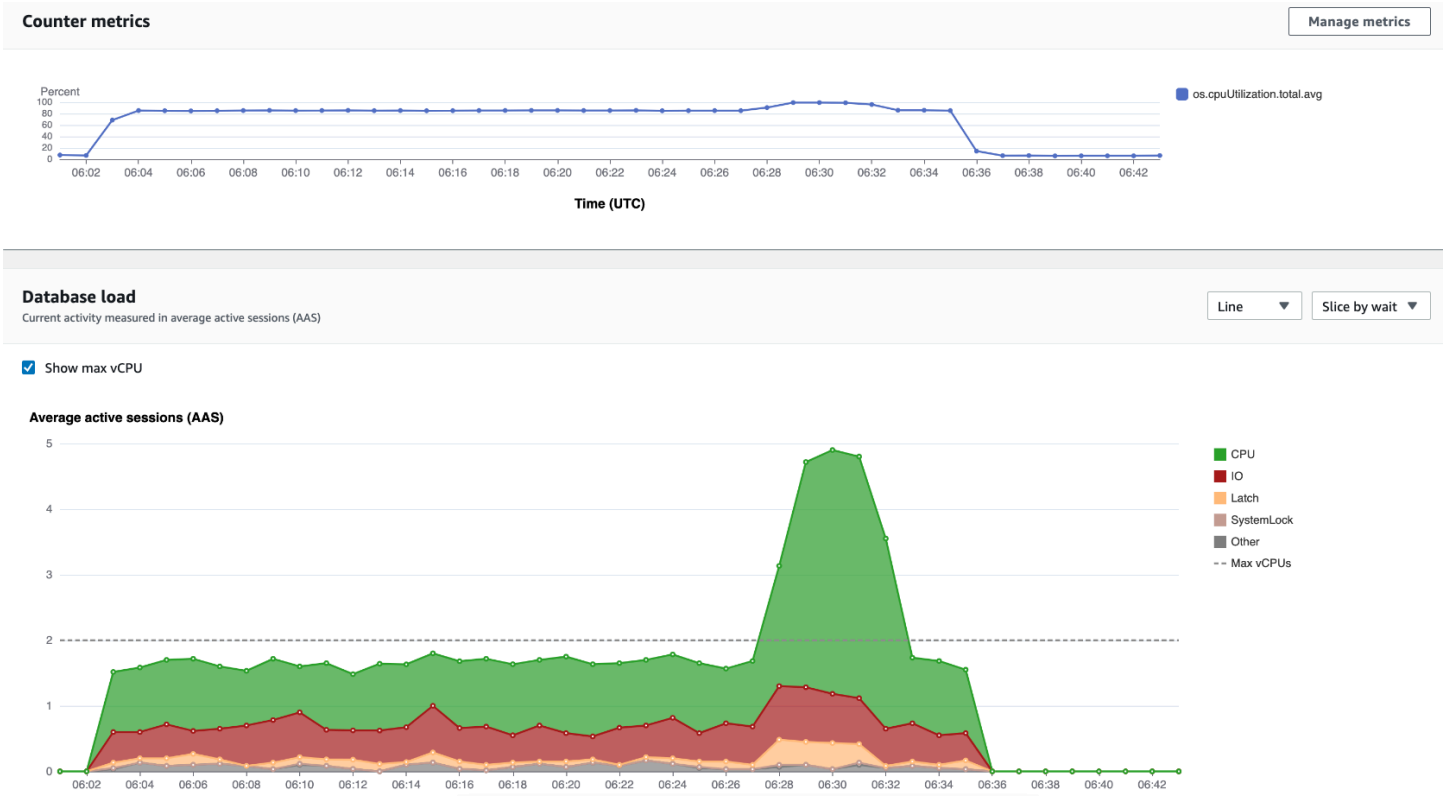
```
aws pi help
```

Falls Sie das nicht AWS CLI installiert haben, finden Sie unter Installation der AWS Befehlszeilenschnittstelle im AWS CLI Benutzerhandbuch weitere Informationen zur Installation.

Abrufen von Zeitreihenmetriken

Mit der `GetResourceMetrics`-Operation werden ein oder mehrere Zeitreihenmetriken aus den Performance Insights-Daten abgerufen. Für `GetResourceMetrics` ist eine Metrik und ein Zeitraum erforderlich, damit eine Antwort mit einer Liste von Datenpunkten zurückgegeben wird.

Zum Beispiel die AWS Management Console Verwendung, `GetResourceMetrics` um die Diagramme „Counter Metrics“ und „Database Load“ auszufüllen, wie in der folgenden Abbildung dargestellt.



Alle von zurückgegebenen Metriken`GetResourceMetrics` sind Standard-Zeitreihenmetriken, mit Ausnahme von`db.load`. Diese Metrik wird im Diagramm Database Load (Datenbanklast)

angezeigt. Die`db.load` Metrik unterscheidet sich von den anderen Zeitreihenmetriken, da Sie sie in Unterkomponenten aufteilen können, die als Dimensionen bezeichnet werden. In der vorherigen Abbildung wird `db.load` unterteilt und nach Wartezuständen gruppiert, aus denen `db.load` besteht.

> **Note**
>
> `GetResourceMetrics` kann auch die `db.sampleload`-Metrik zurückgeben, aber die `db.load`-Metrik ist in den meisten Fällen angemessen.

Informationen zu den Zählermetriken, die von `GetResourceMetrics` zurückgegeben werden, finden Sie unter Performance Insights für Zählermetriken.

Die folgenden Berechnungen werden für die Metriken unterstützt:

- Durchschnitt – Der durchschnittliche Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.avg` an.
- Minimum – Der minimale Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.min` an.
- Maximum – Der maximale Wert für die Metrik über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.max` an.
- Summe – Die Summe der Metrikwerte über einen bestimmten Zeitraum. Fügen Sie dem Metriknamen `.sum` an.
- Beispielanzahl – Die Anzahl, wie oft die Metrik über einen bestimmten Zeitraum erfasst wurde. Fügen Sie dem Metriknamen `.sample_count` an.

Nehmen wir an, dass eine Metrik beispielsweise 300 Sekunden (5 Minuten) lang erfasst wird und dass die Metrik einmal pro Minute erfasst wird. Die Werte für jede Minute sind 1, 2, 3, 4 und 5. In diesem Fall werden die folgenden Berechnungen zurückgegeben:

- Durchschnitt – 3
- Minimum – 1
- Maximum – 5
- Summe – 15
- Beispielanzahl – 5

Hinweise zur Verwendung des `get-resource-metrics` AWS CLI Befehls finden Sie unter `get-resource-metrics`.

Geben Sie für die `--metric-queries`-Option eine oder mehrere Abfragen an, um die entsprechenden Ergebnisse zu erhalten. Jede Abfrage besteht aus einem obligatorischen `Metric`- sowie optionalen `GroupBy`- und `Filter`-Parametern. Es folgt ein Beispiel für eine Spezifikation der `--metric-queries`-Option.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

AWS CLI Beispiele für Performance Insights

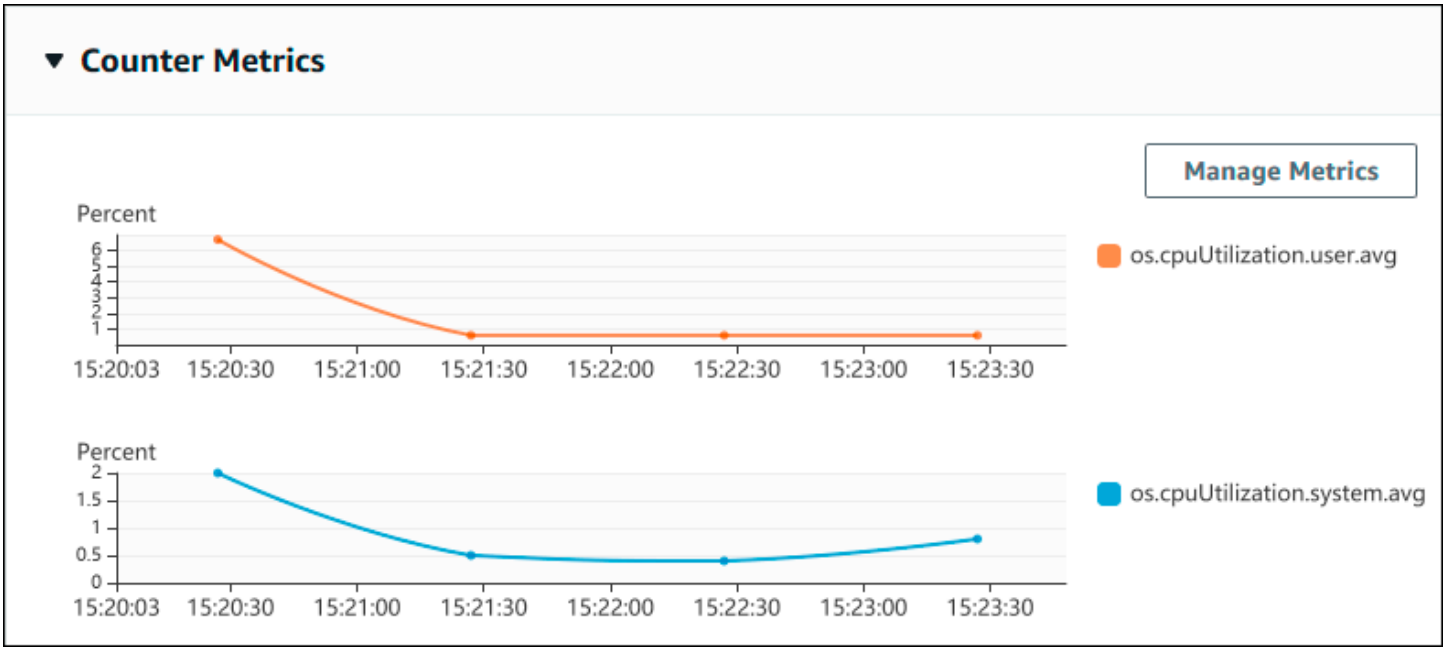
Die folgenden Beispiele zeigen, wie Sie AWS CLI for Performance Insights verwenden können.

Themen

- Abrufen von Zählermetriken
- Der DB-Lastdurchschnitt für die höchsten Wartezeiten wird abgerufen
- Der durchschnittliche DB-Ladestand für die oberste Abfrage wird abgerufen
- Abrufen des nach Query gefilterten DB-Ladedurchschnitts

Abrufen von Zählermetriken

Der folgende Screenshot zeigt zwei Zählermetriken-Diagramme in der AWS Management Console.



Das folgende Beispiel veranschaulicht, wie die Daten, die die AWS Management Console zum Erstellen der beiden Zählermetriken-Diagramme verwendet, gesammelt werden.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Sie können einen Befehl besser lesbar gestalten, indem Sie eine Datei für die Option `--metrics-query` angeben. Im folgenden Beispiel wird eine Datei namens query.json für die Option verwendet. Die Datei enthält Folgendes.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Führen Sie den folgenden Befehl aus, um die Datei zu verwenden.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

Das vorige Beispiel gibt die folgenden Werte für die Optionen an:

- `--service-type`— DOCDB für Amazon DocumentDB
- `--identifier` – Die Ressource-ID für die DB-Instance
- `--start-time` und `--end-time` – Die ISO 8601-Werte `DateTime` für den abzufragenden Zeitraum mit mehreren unterstützten Formaten

Der Abfragezeitraum beträgt eine Stunde:

- `--period-in-seconds` – `60` für eine Abfrage pro Minute
- `--metric-queries` – Ein Array mit zwei Abfragen, jeweils für nur eine Metrik.

  Der Metrikname verwendet Punkte, um die Metrik in eine sinnvolle Kategorie einzustufen, wobei das letzte Element eine Funktion ist. Im Beispiel lautet die Funktion `avg` für jede Abfrage. Wie bei Amazon CloudWatch sind die unterstützten Funktionen `min`, `max`, `total`, und `avg`.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": "2022-03-13T08:00:00+00:00",
    "AlignedEndTime": "2022-03-13T09:00:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {
            "Key": {
                "Metric": "os.cpuUtilization.user.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-03-13T08:01:00+00:00", //Minute1
                    "Value": 3.6
                },
                {
                    "Timestamp": "2022-03-13T08:02:00+00:00", //Minute2
                    "Value": 2.6
                },
                //.... 60 datapoints for the os.cpuUtilization.user.avg metric
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-03-13T08:01:00+00:00",
                    "Value": 92.7
                },
                {
                    "Timestamp": "2022-03-13T08:02:00+00:00",
                    "Value": 93.7
```

```
                },
                //.... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

Die Antwort enthält Werte für `Identifier`, `AlignedStartTime` und `AlignedEndTime`. Bei einem `--period-in-seconds`-Wert von `60` wurden Start- und Endzeiten auf die Minute ausgerichtet. Wenn der `--period-in-seconds`-Wert `3600` lautet, werden Start- und Endzeiten auf die Stunde ausgerichtet.

Die `MetricList` in der Antwort enthält eine Reihe von Einträgen, und zwar jeweils mit einem `Key`- und einem `DataPoints`-Eintrag. Jeder `DataPoint` verfügt über einen `Timestamp` und einen `Value`. Jede `Datapoints`-Liste enthält 60 Datenpunkte, da die Abfragen eine Stunde lang jede Minute Daten abfragen, und zwar mit den Werten `Timestamp1/Minute1`, `Timestamp2/Minute2` usw. bis `Timestamp60/Minute60`.

Da sich die Abfrage auf zwei verschiedene Zählermetriken bezieht, enthält die -Antwort zwei Element `MetricList`.

Der DB-Lastdurchschnitt für die höchsten Wartezeiten wird abgerufen

Das folgende Beispiel ist dieselbe Abfrage, die AWS Management Console verwendet wird, um ein gestapeltes Flächenliniendiagramm zu generieren. In diesem Beispiel wird der Wert `db.load.avg` für die letzte Stunde abgerufen, wobei die Auslastung nach den sieben höchsten Wartezuständen aufgeteilt wird. Der Befehl ist mit dem Befehl unter identisc Abrufen von Zählermetriken. Die Datei query.json enthält hingegen Folgendes.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 7 }
    }
]
```

Führen Sie den folgenden Befehl aus.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
```

```
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
   --end-time   2022-03-13T9:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

Das Beispiel gibt die Metrik für `db.load.avg` und a `GroupBy` der sieben wichtigsten Wartezustände an. Einzelheiten zu gültigen Werten für dieses Beispiel finden Sie DimensionGroupin der Performance Insights API-Referenz.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": "2022-04-04T06:00:00+00:00",
    "AlignedEndTime": "2022-04-04T06:15:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {//A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
                    "Value": 0.0
                },
                {
                    "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
```

```
                    "Value": 0.0
                },
                //... 60 datapoints for the total db.load.avg key
                ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_state.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": "2022-04-04T06:01:00+00:00",//Minute1
                    "Value": 0.0
                },
                {
                    "Timestamp": "2022-04-04T06:02:00+00:00",//Minute2
                    "Value": 0.0
                },
                //... 60 datapoints for the CPU key
            ]
        },//... In total we have 3 key/datapoints entries, 1) total, 2-3) Top Wait
 States
    ] //end of MetricList
} //end of response
```

In dieser Antwort gibt es drei Einträge in der`MetricList`. Es gibt einen Eintrag für die Gesamtzahl `db.load.avg` und jeweils drei Einträge für die `db.load.avg` Aufteilung nach einem der drei höchsten Wartezustände. Da es eine Gruppierungsdimension gab (im Gegensatz zum ersten Beispiel), muss es für jede Gruppierung der Metrik einen Schlüssel geben. Für jede Metrik kann nicht nur ein Schlüssel vorhanden sein, wie im Anwendungsfall der Basiszählermetrik.

Der durchschnittliche DB-Ladestand für die oberste Abfrage wird abgerufen

Das folgende Beispiel `db.wait_state` gruppiert nach den 10 wichtigsten Abfrageanweisungen. Es gibt zwei verschiedene Gruppen für Abfrageanweisungen:

- `db.query`— Die vollständige Abfrageanweisung, wie `{"find":"customers","filter": {"FirstName":"Jesse"},"sort":{"key":{"$numberInt":"1"}}}`

- `db.query_tokenized`— Die tokenisierte Abfrageanweisung, wie `{"find":"customers","filter":{"FirstName":"?"},"sort":{"key": {"$numberInt":"?"}},"limit":{"$numberInt":"?"}}`

Bei der Analyse der Datenbankleistung kann es nützlich sein, Abfrageanweisungen, die sich nur durch ihre Parameter unterscheiden, als ein Logikelement zu betrachten. In diesem Fall können Sie `db.query_tokenized` beim Abfragen verwenden. Vor allem, wenn Sie daran interessiert sind`explain()`, ist es manchmal sinnvoller, vollständige Abfrageanweisungen mit Parametern zu untersuchen. Es besteht eine Beziehung zwischen tokenisierten und vollständigen Abfragen, wobei mehrere vollständige Abfragen (untergeordnete Abfragen) unter derselben tokenisierten Abfrage (übergeordnete Abfrage) gruppiert sind.

Der Befehl in diesem Beispiel ähnelt dem Befehl unter Der DB-Lastdurchschnitt für die höchsten Wartezeiten wird abgerufen. Die Datei query.json enthält hingegen Folgendes.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.query_tokenized", "Limit": 10 }
    }
]
```

Im folgenden Beispiel wird verwende `db.query_tokenized`.

Für Linux, macOS oder Unix:

```
aws pi get-resource-metrics \
   --service-type DOCDB \
   --identifier db-ID \
   --start-time 2022-03-13T8:00:00Z \
   --end-time   2022-03-13T9:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Für Windows:

```
aws pi get-resource-metrics ^
   --service-type DOCDB ^
   --identifier db-ID ^
   --start-time 2022-03-13T8:00:00Z ^
```

```
    --end-time   2022-03-13T9:00:00Z  ^
    --period-in-seconds 3600 ^
    --metric-queries file://query.json
```

In diesem Beispiel werden Abfragen über eine Stunde mit einer Minute abgefragt. period-in-seconds

Das Beispiel gibt die Metrik für `db.load.avg` und einen `GroupBy` der sieben höchsten Wartezustände an. Einzelheiten zu gültigen Werten für dieses Beispiel finden Sie DimensionGroupin der Performance Insights API-Referenz.

Die Antwort sieht in etwa so aus:

```
{
    "AlignedStartTime": "2022-04-04T06:00:00+00:00",
    "AlignedEndTime": "2022-04-04T06:15:00+00:00",
    "Identifier": "db-NQF3TTMFQ3GTOKIMJODMC3KQQ4",
    "MetricList": [
        {//A list of key/datapoints
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //... 60 datapoints for the total db.load.avg key
                ]
        },
               {
            "Key": {//Next key are the top tokenized queries
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.query_tokenized.db_id": "pi-1064184600",
                    "db.query_tokenized.id": "77DE8364594EXAMPLE",
                    "db.query_tokenized.statement": "{\"find\":\"customers\",\"filter
\":{\"FirstName\":\"?\"},\"sort\":{\"key\":{\"$numberInt\":\"?\"}},\"limit\"
:{\"$numberInt\":\"?\"},\"$db\":\"myDB\",\"$readPreference\":{\"mode\":\"primary\"}}"
                }
            },
            "DataPoints": [
            //... 60 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized queries, 1 total key
    ] //End of MetricList
} //End of response
```

Diese Antwort enthält 11 Einträge in der Abfrage `MetricList` (insgesamt 1, 10 am häufigsten tokenisierte Abfragen), wobei jeder Eintrag 24 Einträge pro Stunde enthält. `DataPoints`

Bei tokenisierten Abfragen gibt es drei Einträge in jeder Dimensionsliste:

- `db.query_tokenized.statement`— Die tokenisierte Abfrageanweisung.
- `db.query_tokenized.db_id` — Die synthetische ID, die Performance Insights für Sie generiert. In diesem Beispiel wird die synthetische ID `pi-1064184600` zurückgegeben.
- `db.query_tokenized.id` – Die ID der Abfrage innerhalb von Performance-Insights.

  In der AWS Management Console wird diese ID als Support-ID bezeichnet. Es trägt diesen Namen, weil es sich bei der ID um Daten handelt, die der AWS Support untersuchen kann, um Ihnen bei der Behebung eines Problems mit Ihrer Datenbank zu helfen. AWS nimmt die Sicherheit und den Schutz Ihrer Daten sehr ernst und fast alle Daten werden verschlüsselt zusammen mit Ihren gespeichert AWS KMS key. Daher AWS kann niemand im Inneren diese Daten einsehen. Im vorherigen Beispiel wird sowohl `tokenized.statement` als auch `tokenized.db_id` verschlüsselt gespeichert. Wenn Sie ein Problem mit Ihrer Datenbank haben, kann Ihnen der AWS Support unter Angabe der Support-ID weiterhelfen.

Beim Abfragen empfiehlt es sich ggf., eine `Group` in `GroupBy` anzugeben. Für eine präzisere Kontrolle der Daten, die zurückgegeben werden, sollten Sie allerdings die Dimensionsliste angeben. Wenn z. B. lediglich eine `db.query_tokenized.statement` erforderlich ist, kann der query.json-Datei ein `Dimensions`-Attribut hinzugefügt werden.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.query_tokenized",
            "Dimensions":["db.query_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Abrufen des nach Query gefilterten DB-Ladedurchschnitts

Die entsprechende API-Abfrage in diesem Beispiel ähnelt dem Befehl unter Der durchschnittliche DB-Ladestand für die oberste Abfrage wird abgerufen. Die Datei query.json enthält hingegen Folgendes.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_state", "Limit": 5  },
        "Filter": { "db.query_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
]
```

In dieser Antwort werden alle Werte entsprechend dem Beitrag der tokenisierten Abfrage EXAMPLE gefiltert, die in der Datei AKIAIOSFODNN7 query.json angegeben ist. Die Schlüssel haben möglicherweise auch eine andere Reihenfolge als eine Abfrage ohne Filter, da sich die gefilterte Abfrage auf die fünf Wartezustände mit den höchsten Wartezuständen ausgewirkt hat.

CloudWatch Amazon-Metriken für Performance Insights

Performance Insights veröffentlicht automatisch Metriken auf Amazon CloudWatch. Dieselben Daten können von Performance Insights abgefragt werden, aber wenn die Metriken vorhanden sind, ist es einfach, Alarme hinzuzufügen CloudWatch . CloudWatch Die Metriken können auch leicht zu vorhandenen CloudWatch-Dashboards hinzugefügt werden.

Metrik	Beschreibung
DBLoad	Die Anzahl der aktiven Sitzungen für Amazon DocumentDB. In der Regel sind Sie an den Daten für die durchschnittliche Anzahl der aktiven Sitzungen interessiert. Diese Daten werden in Performance Insights als abgefrag `db.load.avg` .
DBLoadCPU	Die Anzahl der aktiven Sitzungen, bei denen der Wartestatustyp CPU ist. In Performan ce Insights werden diese Daten abgefragt als`db.load.avg` , gefiltert nach dem Wartestatustyp. `CPU`
DBLoadKeine CPU	Die Anzahl der aktiven Sitzungen, bei denen der Wartestatustyp nicht CPU ist.

> **Note**
>
> Diese Metriken werden CloudWatch nur veröffentlicht, wenn die DB-Instance ausgelastet ist.

Sie können diese Metriken mithilfe der CloudWatch Konsole AWS CLI, der oder der CloudWatch API untersuchen.

Sie können beispielsweise die Statistiken für die `DBLoad` Metrik abrufen, indem Sie den get-metric-statisticsBefehl ausführen.

```
aws cloudwatch get-metric-statistics \
    --region ap-south-1 \
    --namespace AWS/DocDB \
    --metric-name DBLoad  \
    --period 360 \
    --statistics Average \
    --start-time 2022-03-14T8:00:00Z \
    --end-time 2022-03-14T9:00:00Z \
    --dimensions Name=DBInstanceIdentifier,Value=documentdbinstance
```

Dieses Beispiel generiert eine Ausgabe wie die folgende.

```
{
    "Datapoints": [
        {
            "Timestamp": "2022-03-14T08:42:00Z",
            "Average": 1.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:24:00Z",
            "Average": 2.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:54:00Z",
            "Average": 6.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:36:00Z",
```

```
            "Average": 5.7,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:06:00Z",
            "Average": 4.0,
            "Unit": "None"
        },
        {
            "Timestamp": "2022-03-14T08:00:00Z",
            "Average": 5.2,
            "Unit": "None"
        }
    ],
    "Label": "DBLoad"
}
```

Sie können die mathematische `DB_PERF_INSIGHTS` Metrikfunktion in der CloudWatch Konsole verwenden, um Zählermetriken von Amazon DocumentDB Performance Insights abzufragen. Die `DB_PERF_INSIGHTS` Funktion beinhaltet auch die `DBLoad` Metrik in Intervallen unter einer Minute. Sie können CloudWatch Alarme für diese Messwerte einrichten. Weitere Informationen zum Erstellen eines Alarms finden Sie unter Erstellen eines Alarms für Performance Insights Insights-Zählermetriken aus einer AWS Datenbank.

Weitere Informationen zu CloudWatch finden Sie unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch.

Performance Insights für Zählermetriken

Zählermetriken sind Betriebssystemmetriken im Performance Insights Insights-Dashboard. Um Leistungsprobleme zu identifizieren und zu analysieren, können Sie Zählermetriken mit der DB-Last korrelieren.

Performance Insights-Betriebssystemzähler

Die folgenden Betriebssystemindikatoren sind mit Amazon DocumentDB Performance Insights verfügbar.

Zähler	Typ	Metrik
Aktiv	memory	os.memory.active
buffers	memory	os.memory.buffers
cached	memory	os.memory.cached
dirty	memory	os.memory.dirty
free	memory	os.memory.free
inactive	memory	os.memory.inactive
mapped	memory	os.memory.mapped
pageTables	memory	os.memory.pageTables
slab	memory	os.memory.slab
total	memory	os.memory.total
writeback	memory	os.memory.writeback
idle	cpuUtilization	os.cpuUtilization.idle
system	cpuUtilization	os.cpuUtilization.system
total	cpuUtilization	os.cpuUtilization.total
user	cpuUtilization	os.cpuUtilization.user
wait	cpuUtilization	os.cpuUtilization.wait
one	loadAverageMinute	OS. loadAverageMinute. eins
fifteen	loadAverageMinute	os. loadAverageMinute. fünfzehn
fünf	loadAverageMinute	os. loadAverageMinute. fünf
cached	swap	os.swap.cached

Zähler	Typ	Metrik
free	swap	os.swap.free
in	swap	os.swap.in
out	swap	os.swap.out
total	swap	os.swap.total
rx	network	os.network.rx
tx	network	os.network.tx
Zahl VCPUs	general	os.general.num VCPUs

Zero-ETL-Integration mit Amazon Service OpenSearch

Themen

- Amazon OpenSearch Service als Ziel
- Einschränkungen

Amazon OpenSearch Service als Ziel

OpenSearch Die Serviceintegration mit Amazon DocumentDB ermöglicht es Ihnen, Volllast- und Änderungsdatenereignisse an OpenSearch Domains zu streamen. Die Aufnahmeinfrastruktur wird als OpenSearch Ingestion-Pipelines gehostet und bietet einen umfassenden Mechanismus mit geringer Latenz, um kontinuierlich Daten aus Amazon DocumentDB-Sammlungen zu streamen.

Bei Volllast extrahiert die Zero-ETL-Integration zunächst historische Volllastdaten, um sie mithilfe einer Aufnahme-Pipeline zu extrahieren. OpenSearch Sobald die Volllastdaten aufgenommen wurden, beginnen die OpenSearch Aufnahme-Pipelines mit dem Lesen von Daten aus Amazon DocumentDB-Change-Streams und catch schließlich auf, um die Datenkonsistenz zwischen Amazon DocumentDB und nahezu in Echtzeit aufrechtzuerhalten. OpenSearch OpenSearch speichert Dokumente in Indizes. Eingehende Daten aus Amazon DocumentDB-Sammlungen können entweder an einen Index gesendet oder in verschiedene Indizes partitioniert werden. Ingestion-Pipelines synchronisieren alle Erstellungs-, Aktualisierungs- und Löschereignisse in einer Amazon DocumentDB-Sammlung als entsprechendes Erstellen, Aktualisieren und Löschen von OpenSearch Dokumenten, um beide Datensysteme synchron zu halten. Aufnahme-Pipelines können so konfiguriert werden, dass sie Daten aus einer Sammlung lesen und in einen Index schreiben oder Daten aus einer Sammlung lesen und bedingt an mehrere Indizes weiterleiten.

Aufnahme-Pipelines können so konfiguriert werden, dass sie Daten von Amazon DocumentDB zu Amazon Service streamen, indem Sie: OpenSearch

- Nur Volllast
- Stream-Change-Stream-Ereignisse von Amazon DocumentDB ohne Volllast streamen
- Volllast, gefolgt von Change-Streams aus Amazon DocumentDB

Gehen Sie wie folgt vor, um Ihre Aufnahme-Pipeline einzurichten:

Schritt 1: Amazon OpenSearch Service-Domain oder OpenSearch serverlose Sammlung erstellen

Eine Amazon OpenSearch Service-Erfassung mit entsprechenden Berechtigungen zum Lesen von Daten ist erforderlich. Informationen zum Erstellen einer Sammlung finden Sie unter Erste Schritte mit Amazon OpenSearch Service oder Erste Schritte mit Amazon OpenSearch Serverless im Amazon OpenSearch Service Developer Guide. Informationen zum Erstellen einer OpenSearch AIM-Rolle mit den richtigen Berechtigungen für den Zugriff auf Schreibdaten für die Sammlung oder Domain finden Sie unter Amazon Ingestion im Amazon OpenSearch Service Developer Guide.

Schritt 2: Change-Streams im Amazon DocumentDB-Cluster aktivieren

Stellen Sie sicher, dass Change-Streams für die erforderlichen Sammlungen im Amazon DocumentDB-Cluster aktiviert sind. Weitere Informationen finden Sie unter Change-Streams mit Amazon DocumentDB verwenden.

Schritt 3: Richten Sie die Pipeline-Rolle mit Schreibberechtigungen für den Amazon S3 S3-Bucket und die Zieldomäne oder Sammlung ein

Nachdem Sie Ihre Amazon DocumentDB-Sammlung erstellt und den Change-Stream aktiviert haben, richten Sie die Pipeline-Rolle ein, die Sie in Ihrer Pipeline-Konfiguration verwenden möchten, und fügen Sie der Rolle die folgenden Berechtigungen hinzu:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowReadAndWriteToS3ForExport",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket/export/*"
            ]
        }
    ]
```

```
}
```

Damit eine OpenSearch Pipeline Daten in eine Domain schreiben kann, muss die OpenSearch Domain über eine Zugriffsrichtlinie auf Domänenebene verfügen, die der Pipeline-Rolle sts_role_arn den Zugriff darauf ermöglicht. Die folgende Beispielrichtlinie für den Domänenzugriff ermöglicht es der Pipeline-Rolle mit dem Namen`pipeline-role`, die Sie im vorherigen Schritt erstellt haben, Daten in die angegebene Domäne zu schreiben: `ingestion-domain`

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:{region}:{your-account-id}:domain/{domain-name}/*"
    }
  ]
}
```

Schritt 4: Fügen Sie die für die Pipeline-Rolle erforderlichen Berechtigungen hinzu, um X-ENI zu erstellen

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:420497401461:network-interface/*",
```

```
                "arn:aws:ec2:*:420497401461:subnet/*",
                "arn:aws:ec2:*:420497401461:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [ "ec2:CreateTags" ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals": { "aws:RequestTag/OSISManaged": "true" }
            }
        }
    ]
}
```

Schritt 5: Erstellen Sie die Pipeline

Konfigurieren Sie eine OpenSearch Aufnahme-Pipeline mit Amazon DocumentDB als Quelle. Bei dieser Beispiel-Pipeline-Konfiguration wird davon ausgegangen, dass ein Mechanismus zum Abrufen von Change-Streams verwendet wird. Weitere Informationen finden Sie unter Verwenden einer OpenSearch Ingestion-Pipeline mit Amazon DocumentDB im Amazon OpenSearch Service Developer Guide.

Einschränkungen

Die folgenden Einschränkungen gelten für die Amazon DocumentDB OpenSearch DocumentDB-Integration:

- Pro Pipeline wird nur eine Amazon DocumentDB-Sammlung als Quelle unterstützt.

- Die regionsübergreifende Datenaufnahme wird nicht unterstützt. Ihr Amazon DocumentDB-Cluster und Ihre OpenSearch Domain müssen sich in derselben AWS Region befinden.
- Die kontoübergreifende Datenaufnahme wird nicht unterstützt. Ihr Amazon DocumentDB-Cluster und Ihre OpenSearch Ingestion-Pipeline müssen sich im selben Konto befinden. AWS
- Elastische Amazon DocumentDB-Cluster werden nicht unterstützt. Es werden nur auf Amazon DocumentDB DocumentDB-Instanzen basierende Cluster unterstützt.
- Stellen Sie sicher, dass für den Amazon DocumentDB-Cluster die Authentifizierung mithilfe von AWS Geheimnissen aktiviert ist. AWS Geheimnisse sind der einzige unterstützte Authentifizierungsmechanismus.
- Die bestehende Pipeline-Konfiguration kann nicht aktualisiert werden, um Daten aus einem anderen and/or a different collection. To update the database and/or Datenbanksammlungsnamen einer Pipeline aufzunehmen. Sie müssen eine neue Pipeline erstellen.

Entwickeln mit Amazon DocumentDB

In diesen Abschnitten wird die Entwicklung mit Amazon DocumentDB behandelt.

Themen

- Connect zu Amazon DocumentDB her
- Programm mit Amazon DocumentDB

Connect zu Amazon DocumentDB her

Amazon DocumentDB bietet mehrere Verbindungsoptionen, darunter programmatische Verbindungen mithilfe von MongoDB-Treibern, Verbindungen von außerhalb einer VPC über SSH-Tunneling oder AWS VPN sowie beliebte Tools wie Studio 3T und. DataGrip Der Service unterstützt Verbindungen mit Replikatgruppen für hohe Verfügbarkeit. Für Analysen und Berichte können Sie mithilfe von JDBC- oder ODBC-Treibern eine Verbindung herstellen. EC2 Instanzen in derselben VPC können sich direkt mit DocumentDB-Clustern verbinden. Alle Verbindungen erfordern TLS und die richtigen AWS Anmeldeinformationen/Authentifizierung.

Themen

- Programmgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB
- Von außerhalb einer Amazon VPC eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen
- Als Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen
- Von Studio 3T aus eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen
- Connect zu Amazon DocumentDB her mit DataGrip
- Stellen Sie über Amazon eine Verbindung her EC2
- Stellen Sie mithilfe des Amazon DocumentDB DocumentDB-JDBC-Treibers eine Verbindung her
- Stellen Sie mithilfe des Amazon DocumentDB DocumentDB-ODBC-Treibers eine Verbindung her

Programmgesteuertes Herstellen einer Verbindung zu Amazon DocumentDB

Dieser Abschnitt enthält Codebeispiele, die zeigen, wie Sie mithilfe verschiedener Sprachen eine Verbindung zu Amazon DocumentDB (mit MongoDB-Kompatibilität) herstellen können. Die Beispiele

sind in zwei Abschnitte unterteilt: Verbindung mit einem Cluster mit und ohne aktiviertem TLS (Transport Layer Security). Standardmäßig ist TLS auf Amazon DocumentDB-Clustern aktiviert. Sie können TLS jedoch bei Bedarf deaktivieren. Weitere Informationen finden Sie unter Verschlüsseln von Daten während der Übertragung.

Wenn Sie versuchen, von außerhalb der VPC, in der sich Ihr Cluster befindet, eine Verbindung zu Ihrer Amazon DocumentDB herzustellen, lesen Sie bitte Von außerhalb einer Amazon VPC eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen.

Bevor Sie sich mit Ihrem Cluster verbinden, müssen Sie wissen, ob TLS auf dem Cluster aktiviert ist. Der nächste Abschnitt zeigt Ihnen, wie Sie den Wert des Parameters `tls` Ihres Clusters mit der AWS Management Console oder der AWS CLI bestimmen können. Danach können Sie fortfahren, indem Sie das entsprechende Codebeispiel anwenden.

Themen

- Ermitteln Sie den Wert Ihres Parameters tls
- Verbindung bei aktiviertem TLS herstellen
- Verbindung bei deaktiviertem TLS herstellen

Ermitteln Sie den Wert Ihres Parameters **`tls`**

Die Feststellung, ob TLS in Ihrem Cluster aktiviert ist, ist ein zweistufiger Prozess, den Sie entweder mit AWS Management Console oder AWS CLI durchführen können.

1. Bestimmen Sie, welche Parametergruppe Ihren Cluster steuert.

   Using the AWS Management Console

   1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
   2. Wählen Sie im linken Navigationsbereich Cluster aus.
   3. Wählen Sie in der Liste der Cluster den Namen des Clusters aus.
   4. Auf der anschließend angezeigten Seite werden die Details des von Ihnen ausgewählten Clusters angezeigt. Wählen Sie die Registerkarte Configuration aus. Suchen Sie im Abschnitt Konfigurationen und Status den Namen der Parametergruppe unter Cluster-Parametergruppe.

Using the AWS CLI

Der folgende AWS CLI Code bestimmt, welcher Parameter Ihren Cluster steuert. Ersetzen Sie *sample-cluster* durch den Namen Ihres Clusters.

```
aws docdb describe-db-clusters \
    --db-cluster-identifier sample-cluster \
    --query 'DBClusters[*].[DBClusterIdentifier,DBClusterParameterGroup]'
```

Die Ausgabe dieser Operation sieht in etwa wie folgt aus:

```
[
       [
           "sample-cluster",
           "sample-parameter-group"
       ]
]
```

2. Bestimmen Sie den Wert des Parameters **`tls`** in der Parametergruppe Ihres Clusters.

Using the AWS Management Console

1. Wählen Sie im Navigationsbereich Parameter groups (Parametergruppen) aus.
2. Wählen Sie im Fenster Cluster-Parametergruppen den Namen Ihrer Cluster-Parametergruppe aus Schritt 1d aus.
3. Auf der Ergebnisseite werden die Parameter der Cluster-Parametergruppe angezeigt. Hier können Sie den Wert des `tls`-Parameters sehen. Informationen zum Ändern dieses Parameters finden Sie unter Amazon DocumentDB-Cluster-Parametergruppen ändern.

Using the AWS CLI

Sie können den `describe-db-cluster-parameters` AWS CLI Befehl verwenden, um die Details der Parameter in Ihrer Cluster-Parametergruppe anzuzeigen.

- **`--describe-db-cluster-parameters`**— Um alle Parameter innerhalb einer Parametergruppe und ihre Werte aufzulisten.
  - **`--db-cluster-parameter-group name`** – Erforderlich. Der Name Ihrer Cluster-Parametergruppe.

Ersetzen Sie in den folgenden Beispielen jeden *user input placeholder* durch die Informationen Ihres Clusters.

```
aws docdb describe-db-cluster-parameters \
    --db-cluster-parameter-group-name sample-parameter-group
```

Die Ausgabe dieser Operation sieht in etwa wie folgt aus:

```
{
        "Parameters": [
            {
                "ParameterName": "profiler_threshold_ms",
                "ParameterValue": "100",
                "Description": "Operations longer than profiler_threshold_ms
 will be logged",
                "Source": "system",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "50-2147483646",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            },
            {
                "ParameterName": "tls",
                "ParameterValue": "disabled",
                "Description": "Config to enable/disable TLS",
                "Source": "user",
                "ApplyType": "static",
                "DataType": "string",
                "AllowedValues": "disabled,enabled,fips-140-3",
                "IsModifiable": true,
                "ApplyMethod": "pending-reboot"
            }
        ]
}
```

> **Note**
>
> Amazon DocumentDB unterstützt FIPS 140-3-Endpunkte, beginnend mit Amazon DocumentDB 5.0-Clustern (Engine-Version 3.0.3727) in diesen Regionen: ca-central-1, us-west-2, us-east-1, us-east-2, -1, -1. us-gov-east us-gov-west

Nachdem Sie den Wert des `tls`-Parameters bestimmt haben, fahren Sie mit dem Herstellen einer Verbindung zu Ihrem Cluster fort, indem Sie eines der Codebeispiele in den folgenden Abschnitten verwenden.

- Verbindung bei aktiviertem TLS herstellen
- Verbindung bei deaktiviertem TLS herstellen

Verbindung bei aktiviertem TLS herstellen

Um ein Codebeispiel für die programmgesteuerte Verbindung zu einem TLS-fähigen Amazon DocumentDB-Cluster anzuzeigen, wählen Sie die entsprechende Registerkarte für die Sprache, die Sie verwenden möchten.

Um Daten während der Übertragung zu verschlüsseln, laden Sie den öffentlichen Schlüssel für Amazon DocumentDB `global-bundle.pem` mit dem folgenden Vorgang herunter.

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

Wenn Ihre Anwendung unter Microsoft Windows läuft und eine PKCS7 Datei benötigt, können Sie das PKCS7 Zertifikatspaket herunterladen. Dieses Paket enthält sowohl die Zwischen- als auch die Stammzertifikate unter https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b.

Python

Der folgende Code zeigt, wie Sie mit Python eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch die Informationen Ihres Clusters.

```
import pymongo
```

```
import sys

##Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
 and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://sample-user:password@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/?tls=true&tlsCAFile=global-
bundle.pem&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
print(x)

##Close the connection
client.close()
```

Node.js

Der folgende Code zeigt, wie Sie mit Node.js eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Ersetzen Sie im folgenden Beispiel jeden durch die Informationen *user input placeholder* Ihres Clusters.

```
var MongoClient = require('mongodb').MongoClient

//Create a MongoDB client, open a connection to DocDB; as a replica set,
//  and specify the read preference as secondary preferred

var client = MongoClient.connect(
'mongodb://sample-user:password@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
tls=true&replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  tlsCAFile: `global-bundle.pem` //Specify the DocDB; cert
},
function(err, client) {
    if(err)
        throw err;

    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'Amazon DocumentDB'}, function(err, result){
        //Print the result to the screen
        console.log(result);

        //Close the connection
        client.close()
      });
   });
});
```

PHP

Der folgende Code zeigt, wie Sie mit PHP eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch die Informationen Ihres Clusters.

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

$TLS_DIR = "/home/ubuntu/global-bundle.pem";

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoDB\Client("mongodb://sample-user:password@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/?retryWrites=false", ["tls" => "true", "tlsCAFile"
 => $TLS_DIR ]);

//Specify the database and collection to be used
$col = $client->sampledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB'] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

Der folgende Code zeigt, wie Sie mit Go eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

> **Note**
>
> Ab Version 1.2.1 verwendet der MongoDB Go Driver nur das erste CA-Serverzertifikat, das in `sslcertificateauthorityfile` gefunden wurde. Der folgende Beispielcode behebt diese Einschränkung, indem alle in `sslcertificateauthorityfile` gefundenen Serverzertifikate manuell an eine benutzerdefinierte TLS-Konfiguration angehängt werden, die während der Client-Erstellung verwendet wird.

Ersetzen Sie in den folgenden Beispielen jedes Beispiel *`user input placeholder`* durch die Informationen Ihres Clusters.

```
package main

import (
 "context"
 "fmt"
 "log"
 "time"

 "go.mongodb.org/mongo-driver/bson"
 "go.mongodb.org/mongo-driver/mongo"
 "go.mongodb.org/mongo-driver/mongo/options"

 "io/ioutil"
 "crypto/tls"
 "crypto/x509"
 "errors"
)

const (
 // Path to the AWS CA file
 caFilePath = "global-bundle.pem"

 // Timeout operations after N seconds
 connectTimeout  = 5
 queryTimeout    = 30
 username        = "sample-user"
 password        = "password"
 clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

 // Which instances to read from
 readPreference = "secondaryPreferred"

 connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
tls=true&replicaSet=rs0&readpreference=%s"
)

func main() {

 connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
 clusterEndpoint, readPreference)

 tlsConfig, err := getCustomTLSConfig(caFilePath)
 if err != nil {
```

```
 log.Fatalf("Failed getting TLS configuration: %v", err)
}

client, err :=
mongo.NewClient(options.Client().ApplyURI(connectionURI).SetTLSConfig(tlsConfig))
if err != nil {
 log.Fatalf("Failed to create client: %v", err)
}

ctx, cancel := context.WithTimeout(context.Background(),
connectTimeout*time.Second)
defer cancel()

err = client.Connect(ctx)
if err != nil {
 log.Fatalf("Failed to connect to cluster: %v", err)
}

// Force a connection to verify our connection string
err = client.Ping(ctx, nil)
if err != nil {
 log.Fatalf("Failed to ping cluster: %v", err)
}

fmt.Println("Connected to DocumentDB!")

collection := client.Database("sample-database").Collection("sample-collection")

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
 log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

cur, err := collection.Find(ctx, bson.D{})
```

```
 if err != nil {
  log.Fatalf("Failed to run find query: %v", err)
 }
 defer cur.Close(ctx)

 for cur.Next(ctx) {
  var result bson.M
  err := cur.Decode(&result)
  log.Printf("Returned: %v", result)

  if err != nil {
   log.Fatal(err)
  }
 }

 if err := cur.Err(); err != nil {
  log.Fatal(err)
 }

}

func getCustomTLSConfig(caFile string) (*tls.Config, error) {
 tlsConfig := new(tls.Config)
 certs, err := ioutil.ReadFile(caFile)

 if err != nil {
  return tlsConfig, err
 }

 tlsConfig.RootCAs = x509.NewCertPool()
 ok := tlsConfig.RootCAs.AppendCertsFromPEM(certs)

 if !ok {
  return tlsConfig, errors.New("Failed parsing pem file")
 }

 return tlsConfig, nil
```

Java

Wenn Sie von einer Java-Anwendung aus eine Verbindung zu einem TLS-fähigen Amazon DocumentDB-Cluster herstellen, muss Ihr Programm die von AWS Ihnen bereitgestellte

Zertifizierungsstelle (CA) verwenden, um die Verbindung zu validieren. Gehen Sie wie folgt vor, um das Amazon RDS-CA-Zertifikat zu verwenden:

1. Laden Sie die Amazon RDS-CA-Datei von herunter https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem.
2. Erstellen Sie einen Vertrauensspeicher mit dem in der Datei enthaltenen CA-Zertifikat, indem Sie die folgenden Befehle ausführen. Stellen Sie sicher, dass Sie das *truststore-password* in etwas anderes ändern. Wenn Sie auf einen Vertrauensspeicher zugreifen, der sowohl das alte CA-Zertifikat (`rds-ca-2015-root.pem`) als auch das neue CA-Zertifikat (`rds-ca-2019-root.pem`) enthält, können Sie das Zertifikat-Bundle in den Vertrauensspeicher importieren.

   Nachfolgend finden Sie ein Beispiel-Shell-Skript, das das Zertifikatpaket in einen Trust Store auf einem Linux-Betriebssystem importiert. Ersetzen Sie in den folgenden Beispielen jedes *user input placeholder* durch Ihre eigenen Informationen. Insbesondere gilt: Wo immer sich das Beispielverzeichnis *mydir* "" im Skript befindet, ersetzen Sie es durch ein Verzeichnis, das Sie für diese Aufgabe erstellt haben.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=truststore-password

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/
 {split_after=1}{print > "rds-ca-" n ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "
```

```
keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Es folgt ein Beispiel für ein Shell-Skript, das das Zertifikatspaket in einen Vertrauensspeicher unter macOS importiert.

```
mydir=/tmp/certs
truststore=${mydir}/rds-truststore.jks
storepassword=truststore-password

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /
Subject:/; s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -
keystore ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep
 Alias | cut -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword}
 -alias "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print
 "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

3. Verwenden Sie das `keystore` in Ihrem Programm, indem Sie die folgenden Systemeigenschaften in Ihrer Anwendung festlegen, bevor Sie eine Verbindung zum Amazon DocumentDB-Cluster herstellen.

```
javax.net.ssl.trustStore: truststore
javax.net.ssl.trustStorePassword: truststore-password;
```

4. Der folgende Code zeigt, wie Sie mit Java eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

   Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch die Informationen Ihres Clusters.

```
package com.example.documentdb;

import com.mongodb.client.*;
import org.bson.Document;

public final class Test {
    private Test() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
ssl=true&replicaSet=rs0&readpreference=%s";
        String username = "sample-user";
        String password = "password";
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        String truststore = "truststore";
        String truststorePassword = "truststore-password";

        System.setProperty("javax.net.ssl.trustStore", truststore);
        System.setProperty("javax.net.ssl.trustStorePassword",
 truststorePassword);

        MongoClient mongoClient = MongoClients.create(connectionString);
```

```
        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection =
 testDB.getCollection("sample-collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }

    }
}
```

C# / .NET

Der folgende Code zeigt, wie Sie mit C# /.NET eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

Im folgenden Beispiel ersetzen Sie jeden *user input placeholder* durch die Informationen Ihres Clusters.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
using MongoDB.Driver;
using MongoDB.Bson;

namespace DocDB
{
    class Program
    {
        static void Main(string[] args)
```

```
        {
            string template = "mongodb://{0}:{1}@{2}/sampledatabase?
tls=true&replicaSet=rs0&readpreference={3}";
            string username = "sample-user";
            string password = "password";
            string readPreference = "secondaryPreferred";
            string clusterEndpoint="sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
            string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);

            string pathToCAFile = "<PATH/global-bundle.p7b_file>";

            // ADD CA certificate to local trust store
            // DO this once - Maybe when your service starts
            X509Store localTrustStore = new X509Store(StoreName.Root);
            X509Certificate2Collection certificateCollection = new
 X509Certificate2Collection();
            certificateCollection.Import(pathToCAFile);
            try
            {
                localTrustStore.Open(OpenFlags.ReadWrite);
                localTrustStore.AddRange(certificateCollection);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Root certificate import failed: " + ex.Message);
                throw;
            }
            finally
            {
                localTrustStore.Close();
            }

            var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
            var client = new MongoClient(settings);

            var database = client.GetDatabase("sampledatabase");
            var collection =
 database.GetCollection<BsonDocument>("samplecollection");
            var docToInsert = new BsonDocument { { "pi", 3.14159 } };
            collection.InsertOne(docToInsert);
        }
```

```
    }
}
```

MongoDB Shell

Der folgende Code zeigt, wie Sie mit der neuesten Version, Mongosh, oder der vorherigen Mongo-Shell-Version eine Verbindung zu Amazon DocumentDB herstellen und diese abfragen, wenn TLS aktiviert ist.

Stellen Sie mit Mongosh eine Connect zu Amazon DocumentDB her

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Ersetzen Sie in den folgenden Beispielen jedes *user input placeholder* Beispiel durch die Informationen Ihres Clusters.

```
mongosh --tls --host cluster-end-point:27017 --tlsCAFile  global-bundle.pem --
username sample-user --password password --retryWrites false
```

Stellen Sie mit der vorherigen Mongo-Shell-Version eine Connect zu Amazon DocumentDB her

Wenn Sie IAM verwenden, müssen Sie eine frühere Version von Mongo Shell verwenden. Geben Sie eine der folgenden Befehlsoptionen ein:

```
mongo --ssl --host cluster-end-point:27017 --sslCAFile global-bundle.pem --
username sample-user --password password
```

Wenn Sie eine Version gleich oder höher als 4.2 verwenden, verwenden Sie den folgenden Code, um eine Verbindung herzustellen. Wiederholbare Schreibvorgänge werden in Amazon DocumentDB nicht unterstützt. Wenn Sie eine ältere Mongo-Shell (nicht Mongosh) verwenden, nehmen Sie den `retryWrites=false` Befehl nicht in eine Codezeichenfolge auf.

Standardmäßig sind wiederholbare Schreibvorgänge deaktiviert. `retryWrites=false`Das Einschließen kann zu einem Fehler bei normalen Lesebefehlen führen.

```
mongo --tls --host cluster-end-point:27017 --tlsCAFile global-bundle.pem --
username sample-user --password password
```

Testen Sie die Verbindung

1. Fügen Sie ein einzelnes Dokument ein.

   ```
   db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
   ```

2. Suchen Sie das Dokument, das zuvor eingefügt wurde.

   ```
   db.myTestCollection.find({'hello':'Amazon DocumentDB'})
   ```

R

Der folgende Code zeigt, wie Sie mit R mithilfe von mongolite (https://jeroen.github.io/mongolite/) eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

Ersetzen Sie im folgenden Beispiel jede durch die Informationen *user input placeholder* Ihres Clusters.

```
#Include the mongolite library.
library(mongolite)

mongourl <- paste("mongodb://sample-user:password@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/test2?ssl=true&",
          "readPreference=secondaryPreferred&replicaSet=rs0", sep="")

#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
#   set and specify the read preference as secondary preferred
client <-  mongo(url = mongourl, options = ssl_options(weak_cert_validation = F, ca
 ="<PATH/global-bundle.pem>"))

#Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)
```

```
#Find the document that was previously written
client$find()
```

Ruby

Der folgende Code zeigt, wie Sie mit Ruby eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS aktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *`user input placeholder`* durch die Informationen Ihres Clusters.

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
   database: 'test',
   replica_set: 'rs0',
   read: {:secondary_preferred => 1},
   user: 'sample-user',
   password: 'password',
   ssl: true,
   ssl_verify: true,
   ssl_ca_cert: 'PATH/global-bundle.pem',
   retry_writes: false
}


begin
   ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
   client = Mongo::Client.new(client_host, client_options)

   ##Insert a single document
   x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

   ##Find the document that was previously written
   result = client[:test].find()

   #Print the document
   result.each do |document|
      puts JSON.neat_generate(document)
   end
```

```
end

#Close the connection
client.close
```

Verbindung bei deaktiviertem TLS herstellen

Um ein Codebeispiel für die programmgesteuerte Verbindung zu einem TLS-deaktivierten Amazon DocumentDB-Cluster anzuzeigen, wählen Sie die Registerkarte für die Sprache, die Sie verwenden möchten.

Python

Der folgende Code zeigt, wie Sie mithilfe von Python eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch die Informationen Ihres Clusters.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set
 and specify the read preference as secondary preferred

import pymongo
import sys

client = pymongo.MongoClient('mongodb://sample-user:password@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false')

##Specify the database to be used
db = client.sample_database

##Specify the collection to be used
col = db.sample_collection

##Insert a single document
col.insert_one({'hello':'Amazon DocumentDB'})

##Find the document that was previously written
x = col.find_one({'hello':'Amazon DocumentDB'})

##Print the result to the screen
```

```
print(x)

##Close the connection
client.close()
```

Node.js

Der folgende Code zeigt, wie Sie mithilfe von Node.js eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Ersetzen Sie im folgenden Beispiel jeden durch die Informationen *user input placeholder* Ihres Clusters.

```
var MongoClient = require('mongodb').MongoClient;

//Create a MongoDB client, open a connection to Amazon DocumentDB as a replica set,
//  and specify the read preference as secondary preferred
var client = MongoClient.connect(
'mongodb://sample-user:password@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/sample-database?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false',
{
  useNewUrlParser: true
},

function(err, client) {
    if(err)
        throw err;
    //Specify the database to be used
    db = client.db('sample-database');

    //Specify the collection to be used
```

```
    col = db.collection('sample-collection');

    //Insert a single document
    col.insertOne({'hello':'Amazon DocumentDB'}, function(err, result){
      //Find the document that was previously written
      col.findOne({'hello':'Amazon DocumentDB'}, function(err, result){
        //Print the result to the screen
        console.log(result);

        //Close the connection
        client.close()
      });
   });
});
```

PHP

Der folgende Code zeigt, wie Sie mit PHP eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch die Informationen Ihres Clusters.

```
<?php
//Include Composer's autoloader
require 'vendor/autoload.php';

//Create a MongoDB client and open connection to Amazon DocumentDB
$client = new MongoDB\Client("mongodb://sample-user:password@sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017/?retryWrites=false");

//Specify the database and collection to be used
$col = $client->sampledatabase->samplecollection;

//Insert a single document
$result = $col->insertOne( [ 'hello' => 'Amazon DocumentDB'] );

//Find the document that was previously written
$result = $col->findOne(array('hello' => 'Amazon DocumentDB'));

//Print the result to the screen
print_r($result);
?>
```

Go

Der folgende Code zeigt, wie Sie mit Go eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel beide *user input placeholder* durch die Informationen Ihres Clusters.

```
package main

import (
 "context"
 "fmt"
 "log"
 "time"

 "go.mongodb.org/mongo-driver/bson"
 "go.mongodb.org/mongo-driver/mongo"
 "go.mongodb.org/mongo-driver/mongo/options"
)

const (
 // Timeout operations after N seconds
 connectTimeout  = 5
 queryTimeout    = 30
 username        = "sample-user"
 password        = "password"
 clusterEndpoint = "sample-cluster.node.us-east-1.docdb.amazonaws.com:27017"

 // Which instances to read from
 readPreference           = "secondaryPreferred"
 connectionStringTemplate = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s"
)

func main() {

 connectionURI := fmt.Sprintf(connectionStringTemplate, username, password,
 clusterEndpoint, readPreference)

 client, err := mongo.NewClient(options.Client().ApplyURI(connectionURI))
 if err != nil {
  log.Fatalf("Failed to create client: %v", err)
```

```
}

ctx, cancel := context.WithTimeout(context.Background(),
connectTimeout*time.Second)
defer cancel()

err = client.Connect(ctx)
if err != nil {
 log.Fatalf("Failed to connect to cluster: %v", err)
}

// Force a connection to verify our connection string
err = client.Ping(ctx, nil)
if err != nil {
 log.Fatalf("Failed to ping cluster: %v", err)
}

fmt.Println("Connected to DocumentDB!")

collection := client.Database("sample-database").Collection("sample-collection")

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

res, err := collection.InsertOne(ctx, bson.M{"name": "pi", "value": 3.14159})
if err != nil {
 log.Fatalf("Failed to insert document: %v", err)
}

id := res.InsertedID
log.Printf("Inserted document ID: %s", id)

ctx, cancel = context.WithTimeout(context.Background(), queryTimeout*time.Second)
defer cancel()

cur, err := collection.Find(ctx, bson.D{})

if err != nil {
 log.Fatalf("Failed to run find query: %v", err)
}
defer cur.Close(ctx)

for cur.Next(ctx) {
 var result bson.M
```

```
  err := cur.Decode(&result)
  log.Printf("Returned: %v", result)

  if err != nil {
   log.Fatal(err)
  }
 }

 if err := cur.Err(); err != nil {
  log.Fatal(err)
 }

}
```

Java

Der folgende Code zeigt, wie Sie mit Java eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch die Informationen Ihres Clusters.

```
package com.example.documentdb;

import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.ServerAddress;
import com.mongodb.MongoException;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoCollection;
import org.bson.Document;


public final class Main {
    private Main() {
    }
    public static void main(String[] args) {

        String template = "mongodb://%s:%s@%s/sample-database?
replicaSet=rs0&readpreference=%s";
        String username = "sample-user";
        String password = "password";
```

```
        String clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
        String readPreference = "secondaryPreferred";
        String connectionString = String.format(template, username, password,
 clusterEndpoint, readPreference);

        MongoClientURI clientURI = new MongoClientURI(connectionString);
        MongoClient mongoClient = new MongoClient(clientURI);

        MongoDatabase testDB = mongoClient.getDatabase("sample-database");
        MongoCollection<Document> numbersCollection = testDB.getCollection("sample-
collection");

        Document doc = new Document("name", "pi").append("value", 3.14159);
        numbersCollection.insertOne(doc);

        MongoCursor<Document> cursor = numbersCollection.find().iterator();
        try {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        } finally {
            cursor.close();
        }

    }
}
```

C# / .NET

Der folgende Code zeigt, wie Sie mit C# /.NET eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch die Informationen Ihres Clusters.

```
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Security.Cryptography.X509Certificates;
using System.Net.Security;
```

```
using MongoDB.Driver;
using MongoDB.Bson;

namespace CSharpSample
{
   class Program
    {
       static void Main(string[] args)
        {
           string template = "mongodb://{0}:{1}@{2}/sampledatabase?
replicaSet=rs0&readpreference={3}";
           string username = "sample-user";
           string password = "password";
           string clusterEndpoint = "sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017";
           string readPreference = "secondaryPreferred";
           string connectionString = String.Format(template, username, password,
 clusterEndpoint, readPreference);

           var settings = MongoClientSettings.FromUrl(new
 MongoUrl(connectionString));
           var client = new MongoClient(settings);

           var database = client.GetDatabase("sampledatabase");
           var collection =
 database.GetCollection<BsonDocument>("samplecollection");
           var docToInsert = new BsonDocument { { "pi", 3.14159 } };
            collection.InsertOne(docToInsert);
        }
    }
}
```

MongoDB Shell

Der folgende Code zeigt, wie Sie mit der neuesten Version, Mongosh, oder der vorherigen Mongo-Shell-Version eine Verbindung zu Amazon DocumentDB herstellen und diese abfragen, wenn TLS deaktiviert ist.

Stellen Sie mit Mongosh eine Connect zu Amazon DocumentDB her

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Ersetzen Sie in den folgenden Beispielen jedes *user input placeholder* Beispiel durch die Informationen Ihres Clusters.

```
mongosh --host cluster-end-point:27017 --username sample-user --password password --
retryWrites false
```

Stellen Sie mit der vorherigen Mongo-Shell-Version eine Connect zu Amazon DocumentDB her

Wenn Sie IAM verwenden, müssen Sie eine frühere Version von Mongo Shell verwenden. Geben Sie eine der folgenden Befehlsoptionen ein:

```
mongo --host cluster-end-point:27017 --username sample-user --password password
```

Wenn Sie eine Version gleich oder höher als 4.2 verwenden, verwenden Sie den folgenden Code, um eine Verbindung herzustellen. Wiederholbare Schreibvorgänge werden in Amazon DocumentDB nicht unterstützt. Wenn Sie eine ältere Mongo-Shell (nicht Mongosh) verwenden, nehmen Sie den `retryWrites=false` Befehl nicht in eine Codezeichenfolge auf. Standardmäßig sind wiederholbare Schreibvorgänge deaktiviert. `retryWrites=false`Das Einschließen kann zu einem Fehler bei normalen Lesebefehlen führen.

```
mongo --host cluster-end-point:27017 --username sample-user --password password
```

Testen Sie die Verbindung

1. Fügen Sie ein einzelnes Dokument ein.

   ```
   db.myTestCollection.insertOne({'hello':'Amazon DocumentDB'})
   ```

2. Suchen Sie das Dokument, das zuvor eingefügt wurde.

```
db.myTestCollection.find({'hello':'Amazon DocumentDB'})
```

R

Der folgende Code zeigt, wie Sie mit R mithilfe von mongolite (https://jeroen.github.io/mongolite/) eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel jede durch die Informationen *user input placeholder* Ihres Clusters.

```
#Include the mongolite library.
library(mongolite)

#Create a MongoDB client, open a connection to Amazon DocumentDB as a replica
#   set and specify the read preference as secondary preferred
client <- mongo(url = "mongodb://sample-user:password@sample-
cluster.node.us-east-1.docdb.amazonaws.com:27017/sample-database?
readPreference=secondaryPreferred&replicaSet=rs0")

##Insert a single document
str <- c('{"hello" : "Amazon DocumentDB"}')
client$insert(str)

##Find the document that was previously written
client$find()
```

Ruby

Der folgende Code zeigt, wie Sie mit Ruby eine Verbindung zu Amazon DocumentDB herstellen, wenn TLS deaktiviert ist.

Ersetzen Sie im folgenden Beispiel jede *user input placeholder* durch die Informationen Ihres Clusters.

```
require 'mongo'
require 'neatjson'
require 'json'
client_host = 'mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017'
client_options = {
   database: 'test',
   replica_set: 'rs0',
```

```
   read: {:secondary_preferred => 1},
   user: 'sample-user',
   password: 'password',
   retry_writes: false
}

begin
   ##Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
   client = Mongo::Client.new(client_host, client_options)

   ##Insert a single document
   x = client[:test].insert_one({"hello":"Amazon DocumentDB"})

   ##Find the document that was previously written
   result = client[:test].find()

   #Print the document
   result.each do |document|
      puts JSON.neat_generate(document)
   end
end

#Close the connection
client.close
```

Von außerhalb einer Amazon VPC eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen

Amazon DocumentDB-Cluster (mit MongoDB-Kompatibilität) werden in einer Amazon Virtual Private Cloud (Amazon VPC) bereitgestellt. Sie können direkt von EC2 Amazon-Instances oder anderen AWS Services abgerufen werden, die in derselben Amazon VPC bereitgestellt werden. Darüber hinaus kann Amazon DocumentDB über VPC-Peering von EC2 Instances oder anderen AWS Services VPCs in verschiedenen Regionen AWS-Region oder in anderen Regionen aufgerufen werden.

Nehmen wir jedoch an, dass Ihr Anwendungsfall erfordert, dass Sie (oder Ihre Anwendung) von außerhalb der VPC des Clusters auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zugreifen. In diesem Fall können Sie SSH-Tunneling (auch bekannt als Portweiterleitung) verwenden, um auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen.

Die eingehende Diskussion zum Thema SSH-Tunneling geht über den Rahmen dieses Themas hinaus. Weitere Informationen zum SSH-Tunneling finden Sie in den folgenden Ressourcen:

- SSH-Tunnel
- Beispiel zum SSH-Port-Forwarding, insbesondere der Abschnitt Lokales Forwarding.

Um einen SSH-Tunnel zu erstellen, benötigen Sie eine EC2 Amazon-Instance, die in derselben Amazon VPC wie Ihr Amazon DocumentDB-Cluster läuft. Sie können entweder eine bestehende EC2 Instanz in derselben VPC wie Ihr Cluster verwenden oder eine erstellen. Weitere Informationen finden Sie im entsprechenden Thema für Ihr Betriebssystem:

- Erste Schritte mit Amazon EC2 Linux-Instances
- Erste Schritte mit Amazon EC2 Windows-Instances

Normalerweise können Sie mit dem folgenden Befehl eine Verbindung zu einer EC2 Instance herstellen.

```
ssh -i "ec2Access.pem" ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com
```

Falls ja, können Sie einen SSH-Tunnel zum Amazon DocumentDB-Cluster einrichten, `sample-cluster.node.us-east-1.docdb.amazonaws.com` indem Sie den folgenden Befehl auf Ihrem lokalen Computer ausführen. Das `-L`-Flag dient zur Weiterleitung eines lokalen Ports. Wenn Sie einen SSH-Tunnel verwenden, empfehlen wir, dass Sie über den Clusterendpunkt eine Verbindung mit Ihrem Cluster herstellen und nicht versuchen, eine Verbindung im Replikatsatzmodus herzustellen (d. h. `replicaSet=rs0` in der Verbindungszeichenfolge anzugeben), da dies zu einem Fehler führt.

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-east-1.docdb.amazonaws.com:27017
 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -N
```

Nachdem der SSH-Tunnel erstellt wurde, werden alle Befehle, an die Sie ausgeben, an den Amazon DocumentDB-Cluster weitergeleitet, der in der Amazon `sample-cluster` VPC ausgeführt wird. `localhost:27017` Wenn Transport Layer Security (TLS) auf Ihrem Amazon DocumentDB-Cluster aktiviert ist, müssen Sie den öffentlichen Schlüssel für Amazon DocumentDB von herunterladen. https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem Mit dem folgenden Vorgang wird diese Datei heruntergeladen:

```
wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem
```

> **Note**
>
> TLS ist standardmäßig für neue Amazon DocumentDB-Cluster aktiviert. Eine Deaktivierung ist jedoch möglich. Weitere Informationen finden Sie unter TLS-Einstellungen für Amazon DocumentDB-Cluster verwalten.

Verwenden Sie den folgenden Befehl, um von außerhalb der Amazon VPC eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen.

```
mongo --sslAllowInvalidHostnames --ssl --sslCAFile global-bundle.pem --username
 <yourUsername> --password <yourPassword>
```

Als Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen

Wenn Sie mit Amazon DocumentDB (mit MongoDB-Kompatibilität) entwickeln, empfehlen wir, dass Sie eine Verbindung zu Ihrem Cluster als Replikatsatz herstellen und Lesevorgänge mithilfe der integrierten Leseeinstellungen Ihres Treibers an Replikatinstanzen verteilen. In diesem Abschnitt wird genauer beschrieben, was das bedeutet, und es wird beschrieben, wie Sie am Beispiel des SDK für Python eine Verbindung zu Ihrem Amazon DocumentDB-Cluster als Replikatsatz herstellen können.

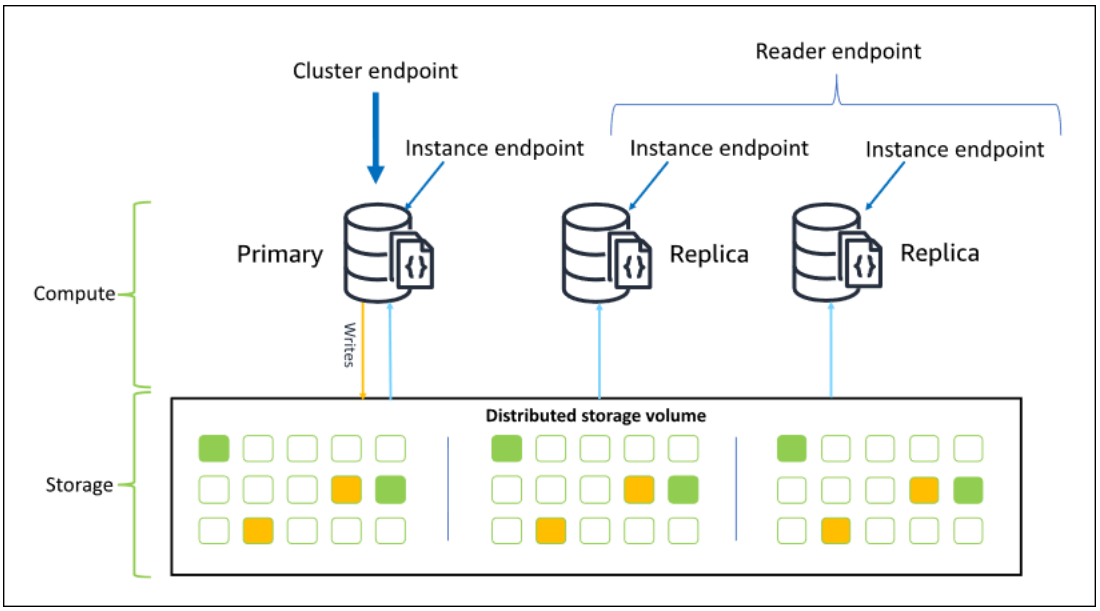
Amazon DocumentDB hat drei Endpunkte, über die Sie eine Verbindung zu Ihrem Cluster herstellen können:

- Cluster-Endpunkt
- Leser-Endpunkt
- Instance-Endpunkte

In den meisten Fällen, wenn Sie eine Verbindung zu Amazon DocumentDB herstellen, empfehlen wir, den Cluster-Endpunkt zu verwenden. Dies ist ein CNAME, der auf die primäre Instance in Ihrem Cluster verweist wie im folgenden Diagramm gezeigt.

Wenn Sie einen SSH-Tunnel verwenden, empfehlen wir, dass Sie über den Clusterendpunkt eine Verbindung mit Ihrem Cluster herstellen und nicht versuchen, eine Verbindung im Replikatsatzmodus herzustellen (d. h. `replicaSet=rs0` in der Verbindungszeichenfolge anzugeben), da dies zu einem Fehler führt.

> **Note**
>
> Weitere Informationen zu Amazon DocumentDB DocumentDB-Endpunkten finden Sie unter. Amazon DocumentDB DocumentDB-Endpunkte



Mithilfe des Cluster-Endpunkts können Sie im Replikatsatzmodus eine Verbindung zu Ihrem Cluster herstellen. Anschließend können Sie die integrierten Treiberfunktionen für Leseeinstellungen verwenden. Im folgenden Beispiel signalisiert die Angabe von `/?replicaSet=rs0` dem SDK, das Sie eine Verbindung als Replikatsatz herstellen möchten. Wenn Sie `/?replicaSet=rs0'` auslassen, leitet der Client alle Anfragen an den Cluster-Endpunkt weiter, d. h. Ihre primäre Instance.

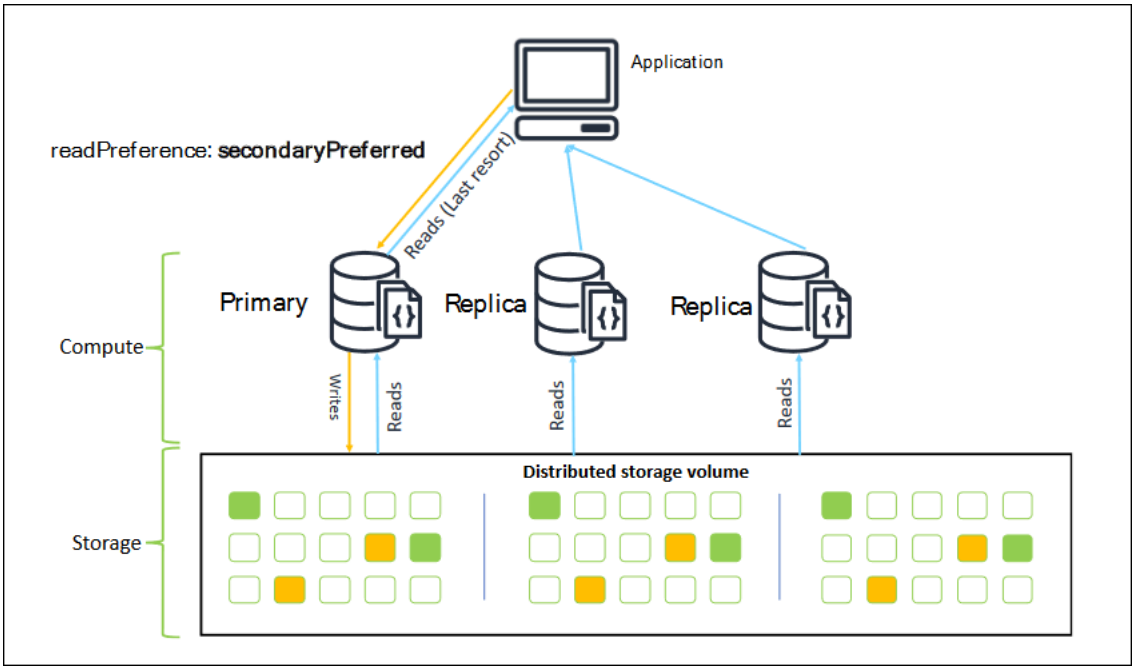
```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ## replica set and specify the read preference as secondary preferred
   client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0')
```

Der Vorteil der Verbindung als Replikatsatz besteht darin, dass Ihr SDK die Cluster-Topographie automatisch erkennt, auch wenn Instances dem Cluster hinzugefügt oder aus dem Cluster entfernt werden. Anschließend können Sie den Cluster effizienter nutzen, indem Sie Leseanforderungen an Ihre Replikat-Instances weiterleiten.

Wenn Sie eine Verbindung als Replikatsatz herstellen, können Sie die `readPreference` für die Verbindung angeben. Wenn Sie die Leseeinstellung `secondaryPreferred` angeben, leitet der

Client Leseabfragen an Ihre Replikate und Schreibabfragen an Ihre primäre Instance weiter (wie im folgenden Diagramm gezeigt). Damit werden Ihre Cluster-Ressourcen besser genutzt. Weitere Informationen finden Sie unter Lesen Sie die Einstellungsoptionen.

```
## Create a MongoDB client, open a connection to Amazon DocumentDB as a
   ##   replica set and specify the read preference as secondary preferred
client = pymongo.MongoClient('mongodb://<user-name>:<password>@mycluster.node.us-
east-1.docdb.amazonaws.com:27017/?replicaSet=rs0&readPreference=secondaryPreferred')
```



Lesevorgänge von Amazon DocumentDB DocumentDB-Replikaten sind letztendlich konsistent. Sie geben die Daten in der gleichen Reihenfolge zurück, in der sie auf dem primären Knoten geschrieben wurden. Die Replikationsverzögerung beträgt häufig weniger als 50 ms. Sie können die Replikatverzögerung für Ihren Cluster mithilfe der CloudWatch Amazon-Metriken `DBInstanceReplicaLag` und `DBClusterReplicaLagMaximum` überwachen. Weitere Informationen finden Sie unter Überwachen von Amazon DocumentDB mit CloudWatch.

Im Gegensatz zur herkömmlichen monolithischen Datenbankarchitektur trennt Amazon DocumentDB Speicher und Datenverarbeitung. Aufgrund dieser modernen Architektur sollten Sie Lesevorgänge auf Replikat-Instances skalieren. Lesevorgänge auf Replikat-Instances blockieren keine Schreibvorgänge, die von der primären Instance repliziert werden. Sie können einem Cluster bis zu 15 Read-Reaplica-Instances hinzufügen und auf Millionen von Lesevorgängen pro Sekunde skalieren.

Der Hauptvorteil der Verbindung als Replikatsatz und der Verteilung von Lesevorgängen an Replikate besteht darin, dass dies die Gesamtzahl der Ressourcen im Cluster erhöht, die für Ihre Anwendung verfügbar sind. Wir empfehlen die Verbindung als Replikatsatz als bewährte Methode. Darüber hinaus empfehlen wir diesen Ansatz besonders für die folgenden Szenarien:

- Sie nutzen nahezu 100 Prozent der CPU-Leistung auf Ihrer primären Instance.
- Das Puffer-Cache-Treffer-Verhältnis ist beinahe Null.
- Sie erreichen die Verbindungs- oder Cursor-Limits für eine einzelne Instance.

Die Aufwärtsskalierung einer Cluster-Instance-Größe ist eine Option und kann in einigen Fällen die beste Möglichkeit für die Skalierung des Clusters darstellen. Sie sollten jedoch auch überlegen, wie Sie die Replikate, die in Ihrem Cluster bereits vorhanden sind, besser nutzen können. So können Sie die Skalierung erhöhen, ohne dass für Sie höhere Kosten für die Verwendung eines größeren Instance-Typs anfallen. Wir empfehlen außerdem, diese Grenzwerte (d. h., und`BufferCacheHitRatio`) mithilfe von CloudWatch Alarmen zu überwachen und Warnmeldungen zu erstellen `CPUUtilization``DatabaseConnections`, damit Sie wissen, wenn eine Ressource stark beansprucht wird.

Weitere Informationen finden Sie unter den folgenden Themen:

- Bewährte Methoden für Amazon DocumentDB
- Amazon DocumentDB Kontingente und Beschränkungen

Verwenden von Clusterverbindungen

Betrachten Sie ein Szenario, in dem alle Verbindungen in Ihrem Cluster genutzt werden. Beispielsweise hat eine `r5.2xlarge`-Instance ein Limit von 4.500 Verbindungen (und 450 offenen Cursors). Wenn Sie einen Amazon DocumentDB-Cluster mit drei Instanzen erstellen und nur über den Cluster-Endpunkt eine Verbindung zur primären Instance herstellen, liegen Ihre Cluster-Grenzwerte für offene Verbindungen und Cursor bei 4.500 bzw. 450. Sie können diese Limits erreichen, wenn Sie Anwendungen mit zahlreichen Workern verwenden, die in Containern gestartet werden. Die Container öffnen mehrere Verbindungen gleichzeitig und sättigen den Cluster.

Stattdessen könnten Sie sich als Replikatsatz mit dem Amazon DocumentDB-Cluster verbinden und Ihre Lesevorgänge an die Replikatinstanzen verteilen. Anschließend könnten Sie die Anzahl der verfügbaren Verbindungen und Cursors im Cluster effektiv auf 13.500 bzw. 1.350 verdreifachen. Das Hinzufügen weiterer Instances zum Cluster erhöht lediglich die Anzahl der Verbindungen und Cursors

für Lese-Workloads. Wenn Sie die Anzahl der Verbindungen für Schreibvorgänge in Ihrem Cluster erhöhen müssen, sollten Sie die Instance-Größe erhöhen.

> **Note**
>
> Die Anzahl der Verbindungen für `large`, `xlarge`, und `2xlarge` Instances erhöht sich mit der Instance-Größe bis zu 4.500. Die maximale Anzahl von Verbindungen pro Instance für `4xlarge` Instances oder größer beträgt 4.500. Weitere Hinweise zu Beschränkungen nach Instance-Typ finden Sie unter Instance-Limits.

In der Regel empfehlen wir das Herstellen von Verbindungen mit Ihrem Cluster unter Verwendung der Leseeinstellung `secondary` nicht. Der Grund hierfür ist, dass die Lesevorgänge fehlschlagen, wenn es keine Replikat-Instances im Cluster gibt. Nehmen wir zum Beispiel an, Sie haben einen Amazon DocumentDB-Cluster mit zwei Instanzen, einem primären und einem Replikat. Wenn die Replikat-Instance ein Problem aufweist, schlagen Leseanforderungen aus einem als `secondary` festgelegten Verbindungspool fehl. Der Vorteil von `secondaryPreferred` besteht darin, dass der Client für Lesevorgänge auf den primären Knoten zurückgreift, wenn er keine geeignete Replikat-Instance für die Verbindung finden kann.

Mehrere Verbindungspools

In einigen Szenarien müssen Lesevorgänge in einer Anwendung read-after-write konsistent sein, sodass sie nur von der primären Instance in Amazon DocumentDB bedient werden können. In diesen Szenarien könnten Sie zwei Client-Verbindungspools erstellen: einen für Schreibvorgänge und einen für Lesevorgänge, die read-after-write Konsistenz erfordern. Ihr Code würde in diesem Fall ungefähr wie folgt aussehen.

```
## Create a MongoDB client,
##    open a connection to Amazon DocumentDB as a replica set and specify the
 readPreference as primary
clientPrimary = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=primary')

## Create a MongoDB client,
##    open a connection to Amazon DocumentDB as a replica set and specify the
 readPreference as secondaryPreferred
```

```
secondaryPreferred = pymongo.MongoClient('mongodb://<user-
name>:<password>@mycluster.node.us-east-1.docdb.amazonaws.com:27017/?
replicaSet=rs0&readPreference=secondaryPreferred')
```

Eine weitere Möglichkeit besteht darin, einen einzelnen Verbindungspool zu erstellen und die Leseeinstellung für eine bestimmte Sammlung zu überschreiben.

```
##Specify the collection and set the read preference level for that collection
col = db.review.with_options(read_preference=ReadPreference.SECONDARY_PREFERRED)
```

Übersicht

Um die Ressourcen im Cluster besser zu nutzen, sollten Sie Verbindungen mit dem Cluster über den Replikatsatzmodus herstellen. Sie können die Zahl der Lesevorgänge für Ihre Anwendung skalieren, indem Sie Ihre Lesevorgänge an die Replikat-Instances verteilen, wenn dies für Ihre Anwendung geeignet ist.

Von Studio 3T aus eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen

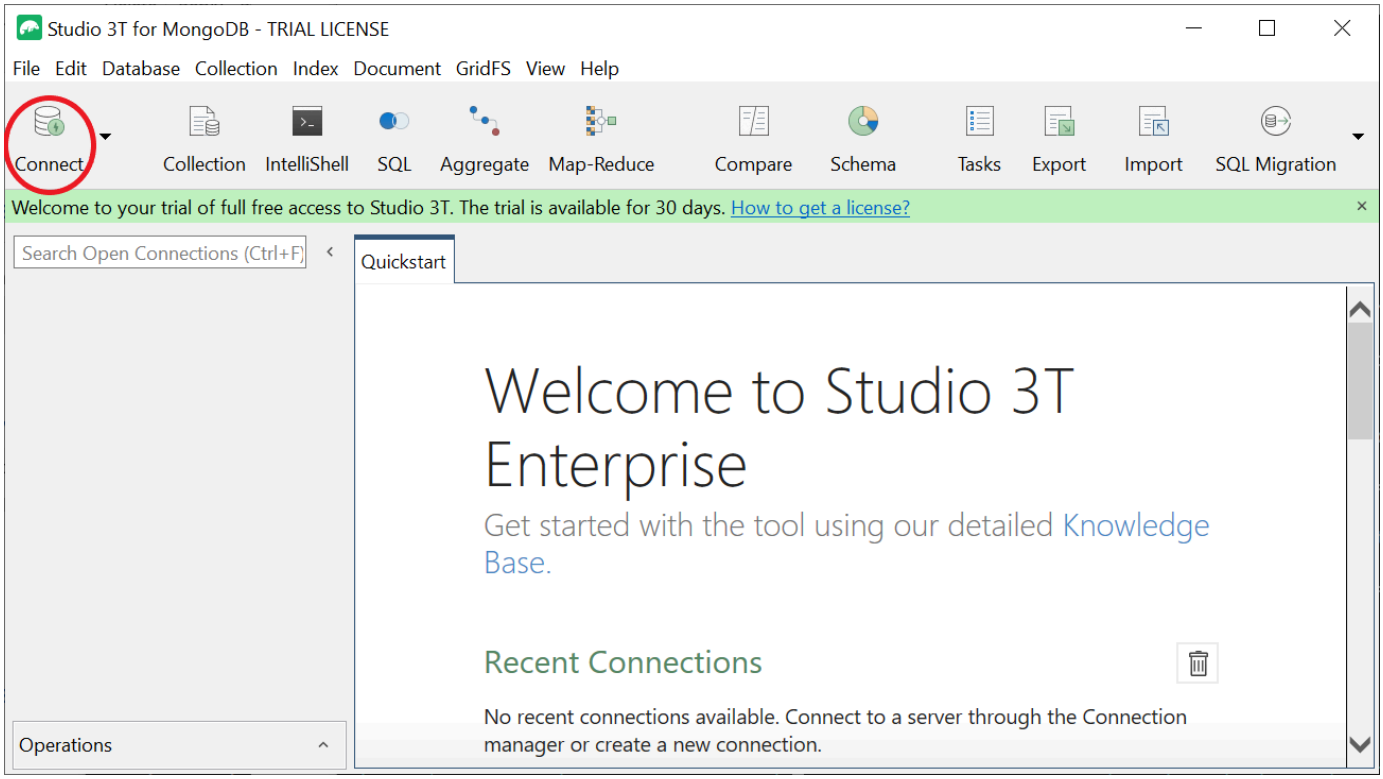
Studio 3T ist eine beliebte GUI und IDE für Entwickler und Dateningenieure, die mit MongoDB arbeiten. Es bietet mehrere leistungsstarke Funktionen Baum-, Tabellen- und JSON-Ansichten Ihrer Daten, eine einfacheimport/export in CSV, JSON, SQL and BSON/mongodump, flexible Abfrageoption, eine visuelle drag-and-drop Benutzeroberfläche, eine integrierte Mongo-Shell mit automatischer Vervollständigung, einen Aggregationspipeline-Editor und SQL-Abfrageunterstützung.

Voraussetzungen

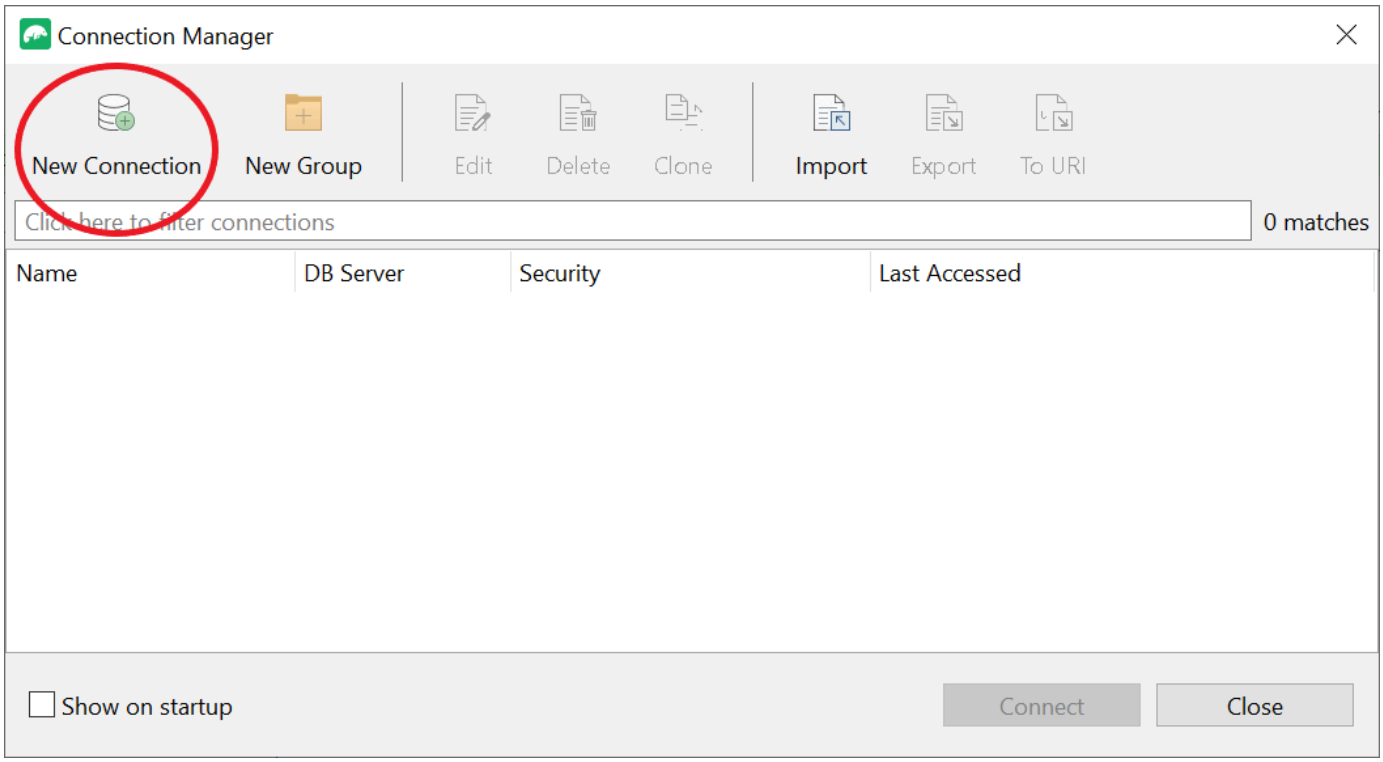
- Wenn Sie noch keinen Amazon DocumentDB-Cluster haben, der Amazon EC2 als Bastion/Jump-Host verwendet, folgen Sie den Anweisungen zur Connect mit Amazon. EC2
- Wenn Sie Studio 3T nicht haben, laden Sie es herunter und installieren Sie es.

Mit Studio 3T Connect

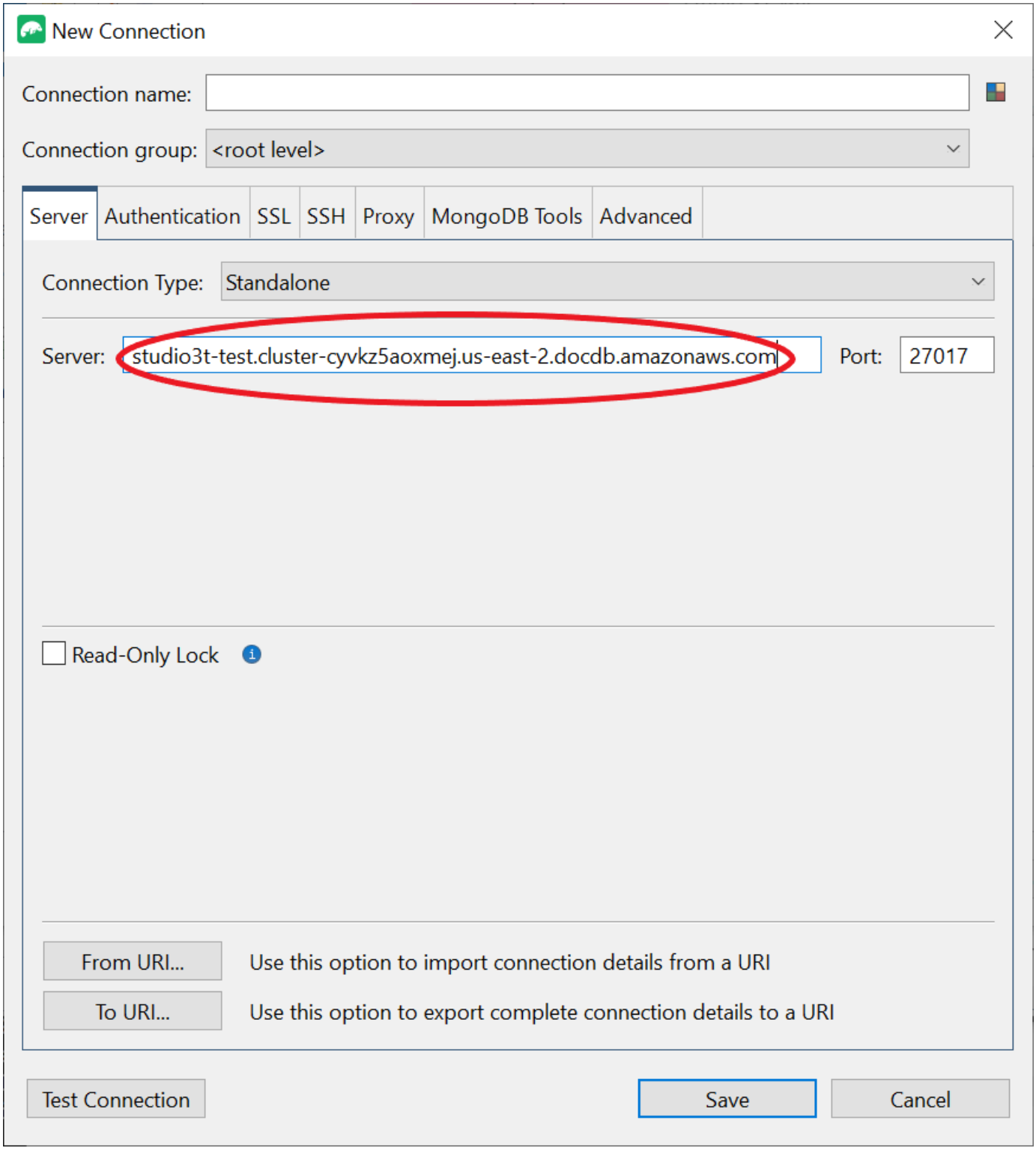
1. Wählen Sie in der oberen linken Ecke der Werkzeugleiste Connect.



2. Wählen Sie in der oberen linken Ecke der Werkzeugleiste Neue Verbindung.

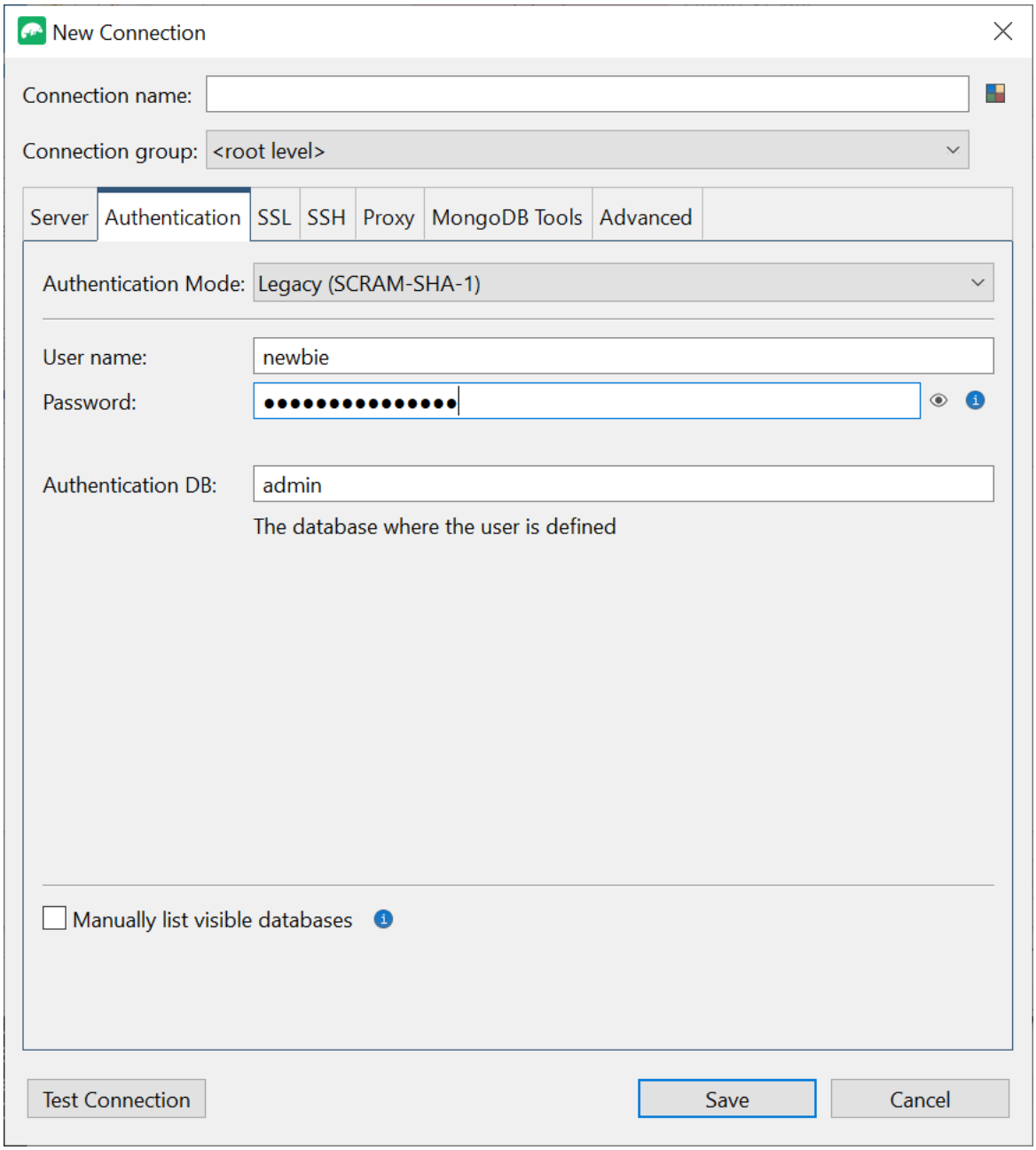


3. Geben Sie auf der Registerkarte Server im Feld Server die Informationen zum Cluster-Endpunkt ein.



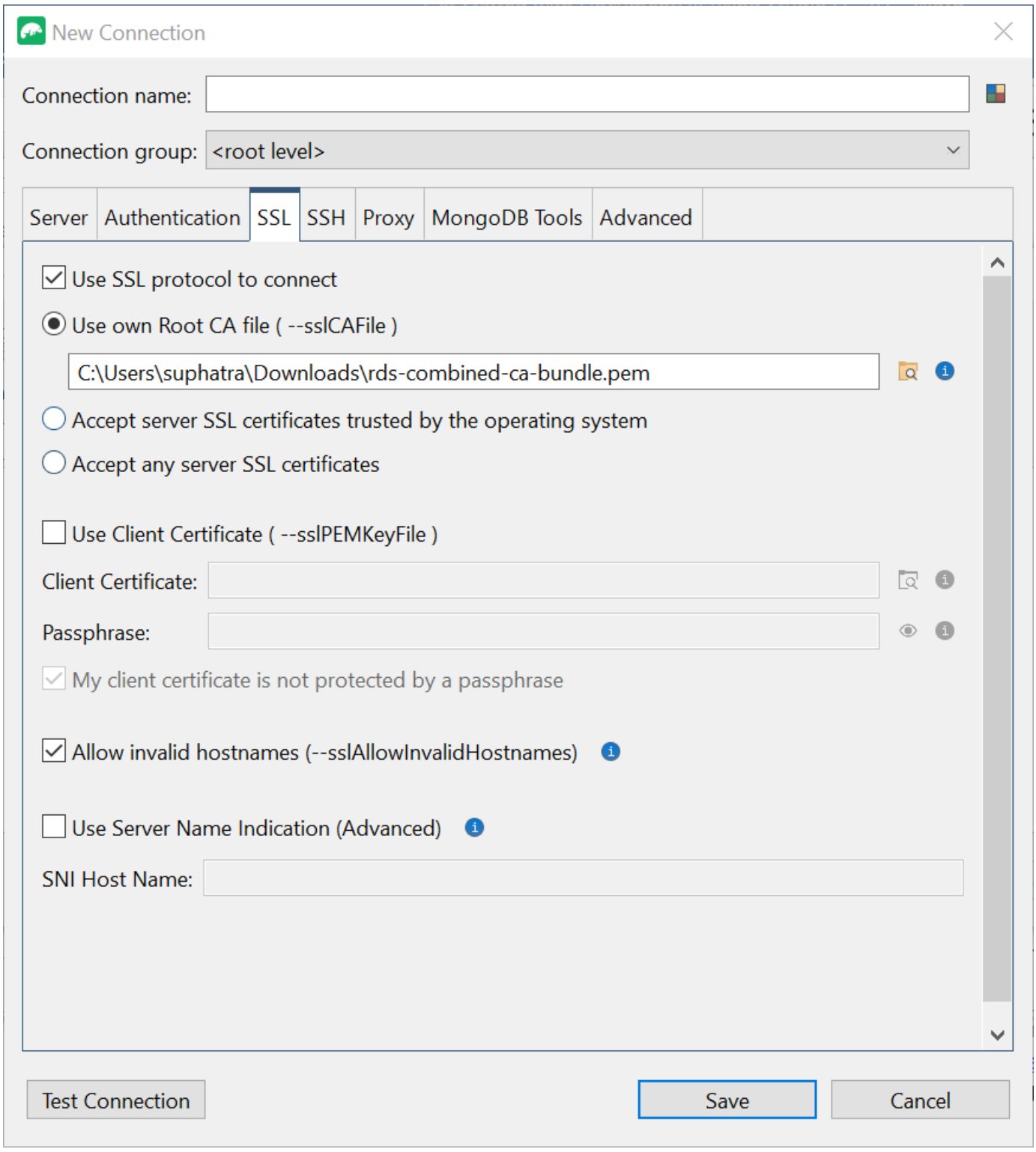
> **Note**
>
> Sie können Ihren Cluster-Endpunkt nicht finden? Folgen Sie einfach den Schritten hier.

4. Wählen Sie die Registerkarte Authentifizierung und wählen Sie Legacy im Drop-down-Menü für Authentifizierungsmodus aus.

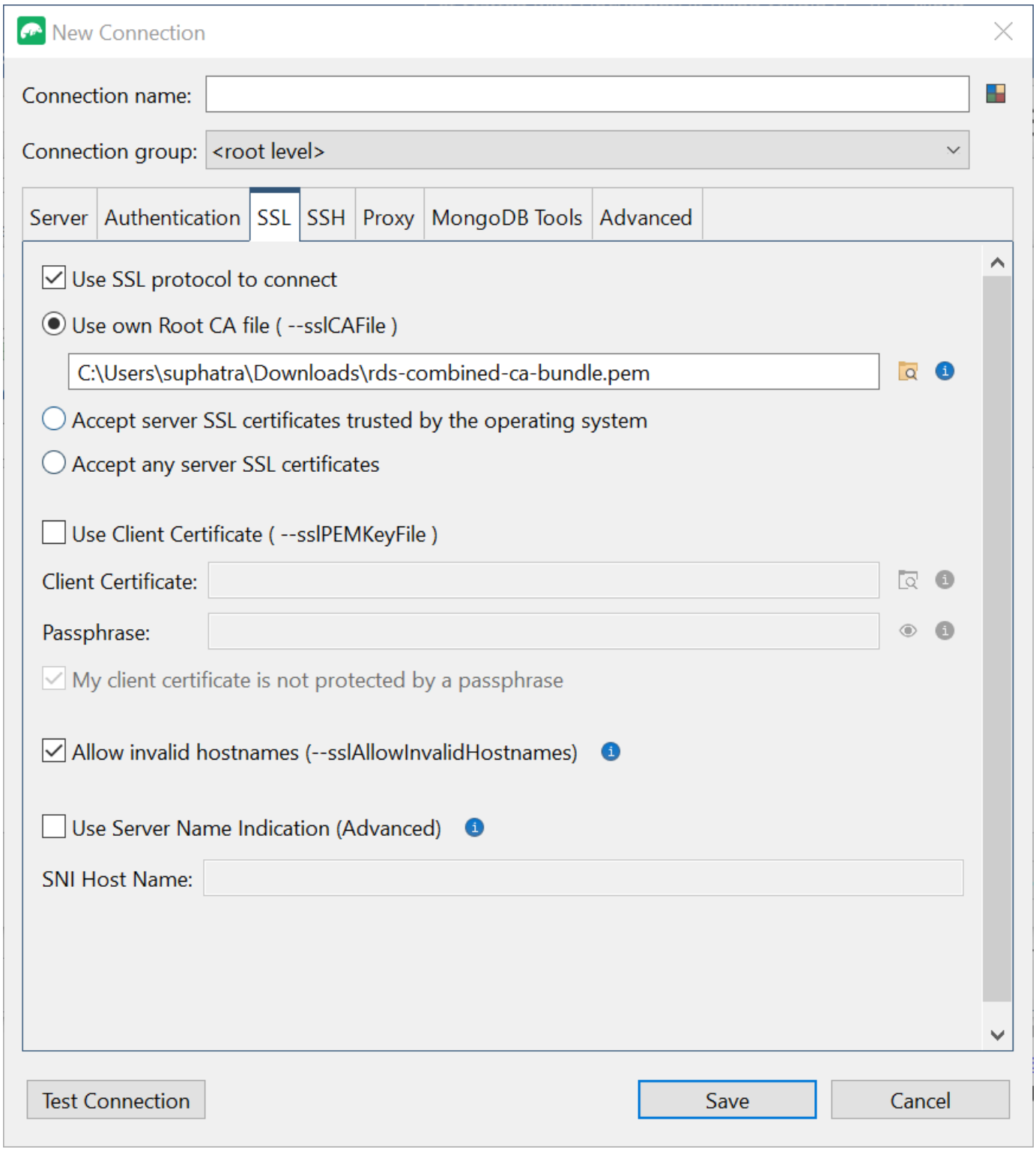


5. Geben Sie Ihren Benutzernamen und Ihre Anmeldeinformationen in die Felder Benutzername und Passwort ein.

6. Wählen Sie die Registerkarte SSL und aktivieren Sie das Kästchen SSL-Protokoll für die Verbindung verwenden.



7. Wählen Sie Eigene Root-CA-Datei verwenden. Fügen Sie dann das Amazon DocumentDB DocumentDB-Zertifikat hinzu (Sie können diesen Schritt überspringen, wenn SSL in Ihrem DocumentDB-Cluster deaktiviert ist). Markieren Sie das Kästchen, um ungültige Hostnamen zuzulassen.



> **Note**
>
> Sie haben das Zertifikat nicht? Sie können es mit dem folgenden Befehl herunterladen:

```
wget https://truststore.pki.rds.amazonaws.com/global/global-
bundle.pem
```

8. Wenn Sie von einem Client-Computer außerhalb der Amazon VPC aus eine Verbindung herstellen, müssen Sie einen SSH-Tunnel erstellen. Sie werden dies auf der Registerkarte SSH tun.

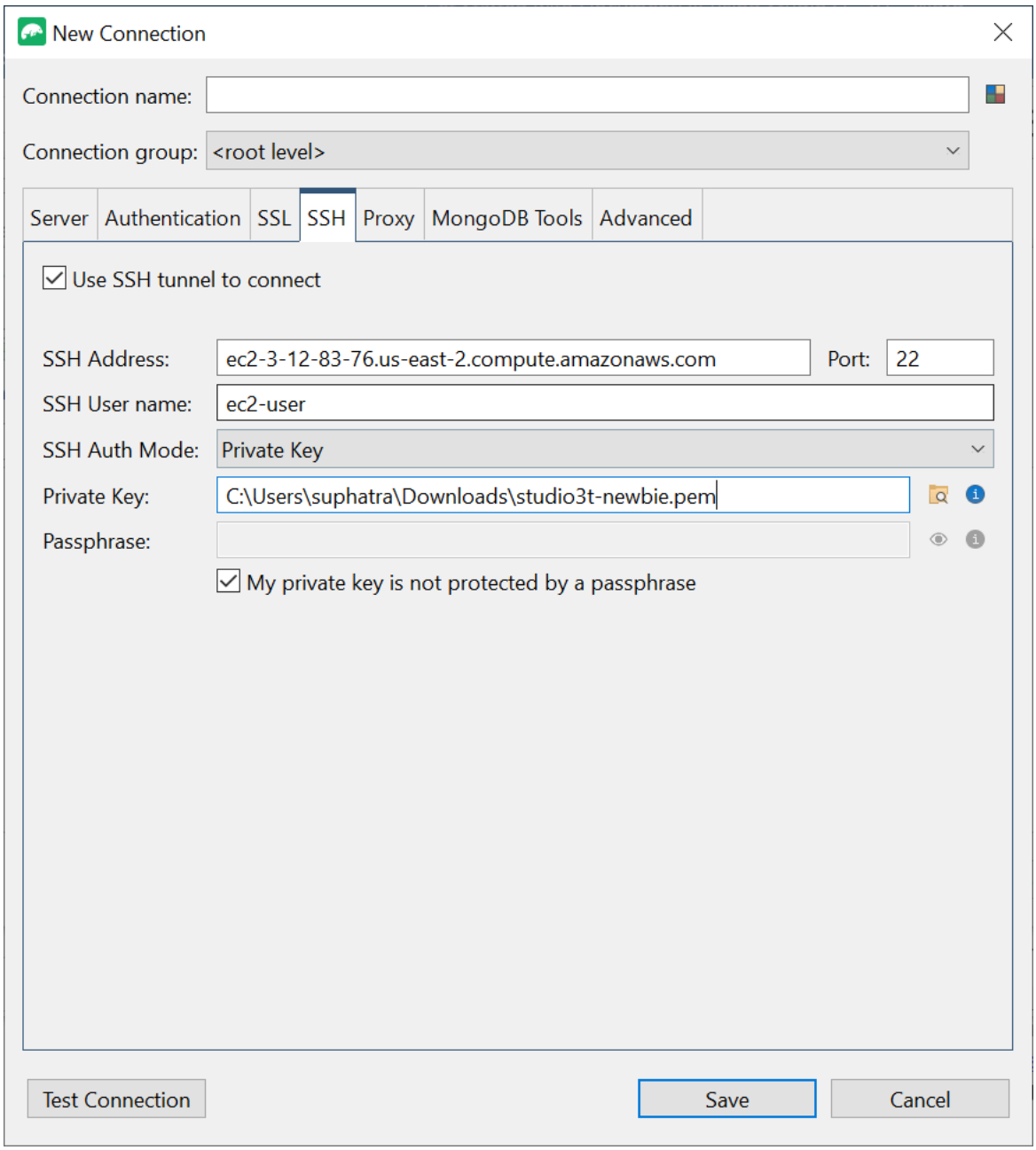
   a. Aktivieren Sie das Kontrollkästchen SSH-Tunnel verwenden und geben Sie die SSH-Adresse in das Feld SSH-Adresse ein. Dies ist Ihre Instanz Public DNS (). IPV4 Sie können diese URL von Ihrer Amazon EC2 Management Console abrufen.

   b. Geben Sie Ihren Nutzernamen ein. Dies ist der Benutzername Ihrer EC2 Amazon-Instance

   c. Wählen Sie für den SSH-Authentifizierungsmodus Private Key aus. Wählen Sie im Feld Privater Schlüssel das Dateifinder-Symbol aus, um den privaten Schlüssel Ihrer EC2 Amazon-Instance zu finden, und wählen Sie ihn aus. Dies ist die PEM-Datei (key pair), die Sie beim Erstellen Ihrer Instance in der Amazon EC2 Console gespeichert haben.

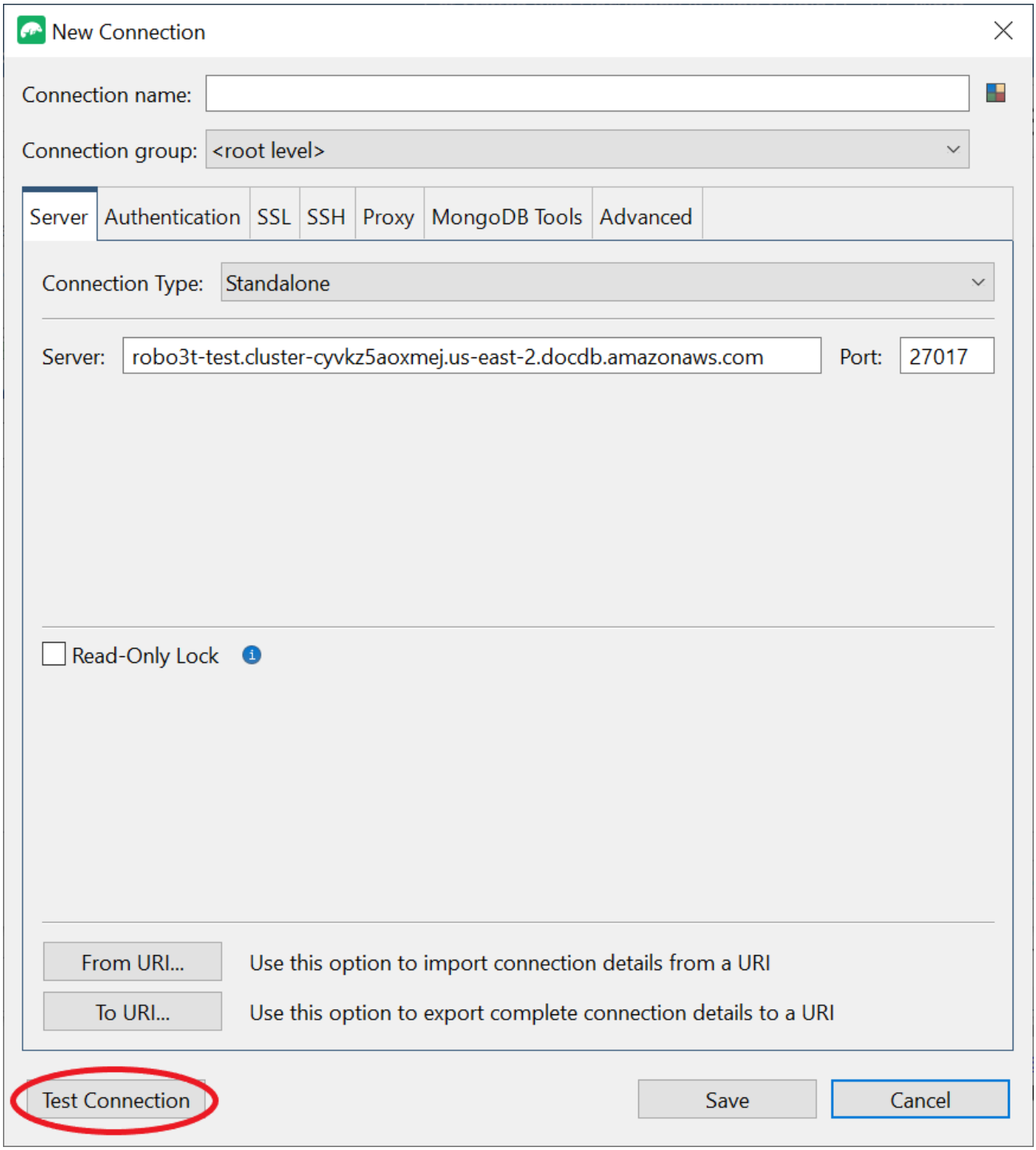
   d. Wenn Sie sich auf einem Linux/macOS-Client-Computer befinden, müssen Sie möglicherweise die Berechtigungen Ihres privaten Schlüssels mit dem folgenden Befehl ändern:

      ```
      chmod 400 /fullPathToYourPemFile/<yourKey>.pem
      ```

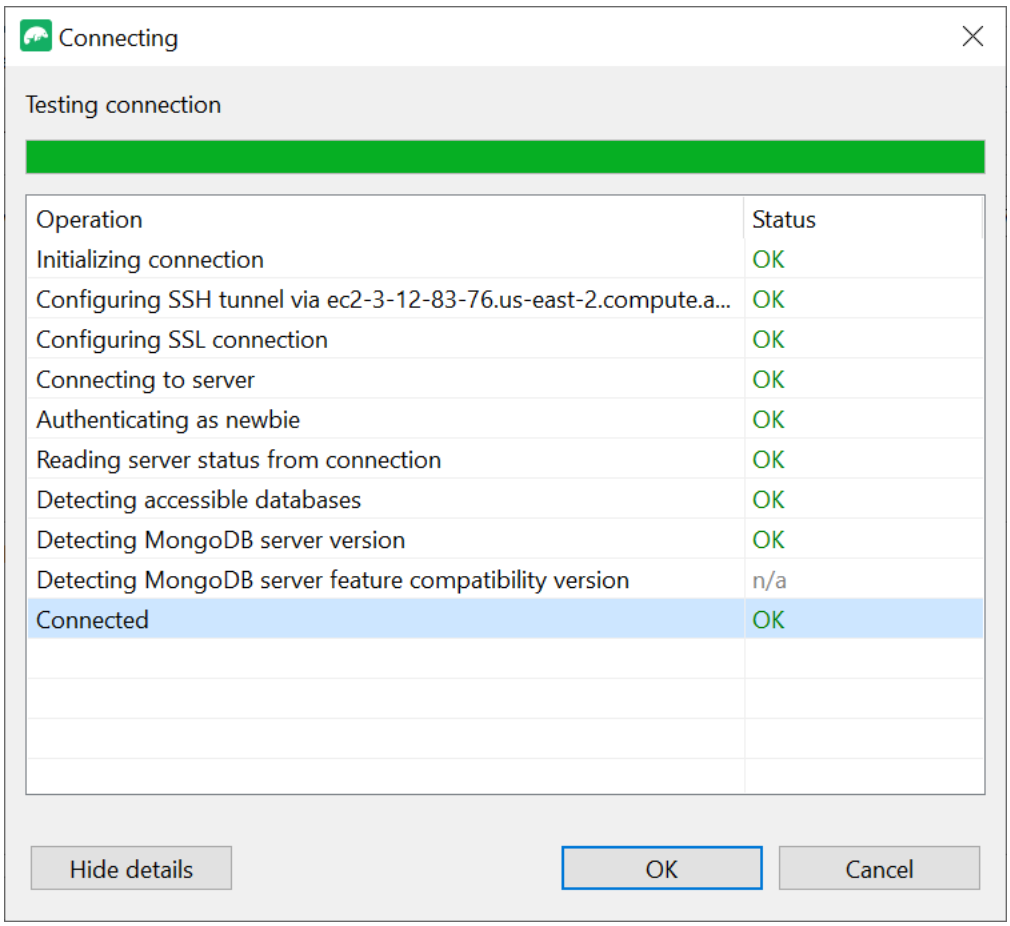


> **Note**
>
> Diese EC2 Amazon-Instance sollte sich in derselben Amazon-VPC und Sicherheitsgruppe wie Ihr DocumentDB-Cluster befinden. Sie können die SSH-Adresse, den Benutzernamen und den privaten Schlüssel von Ihrer Amazon EC2 Management Console abrufen.

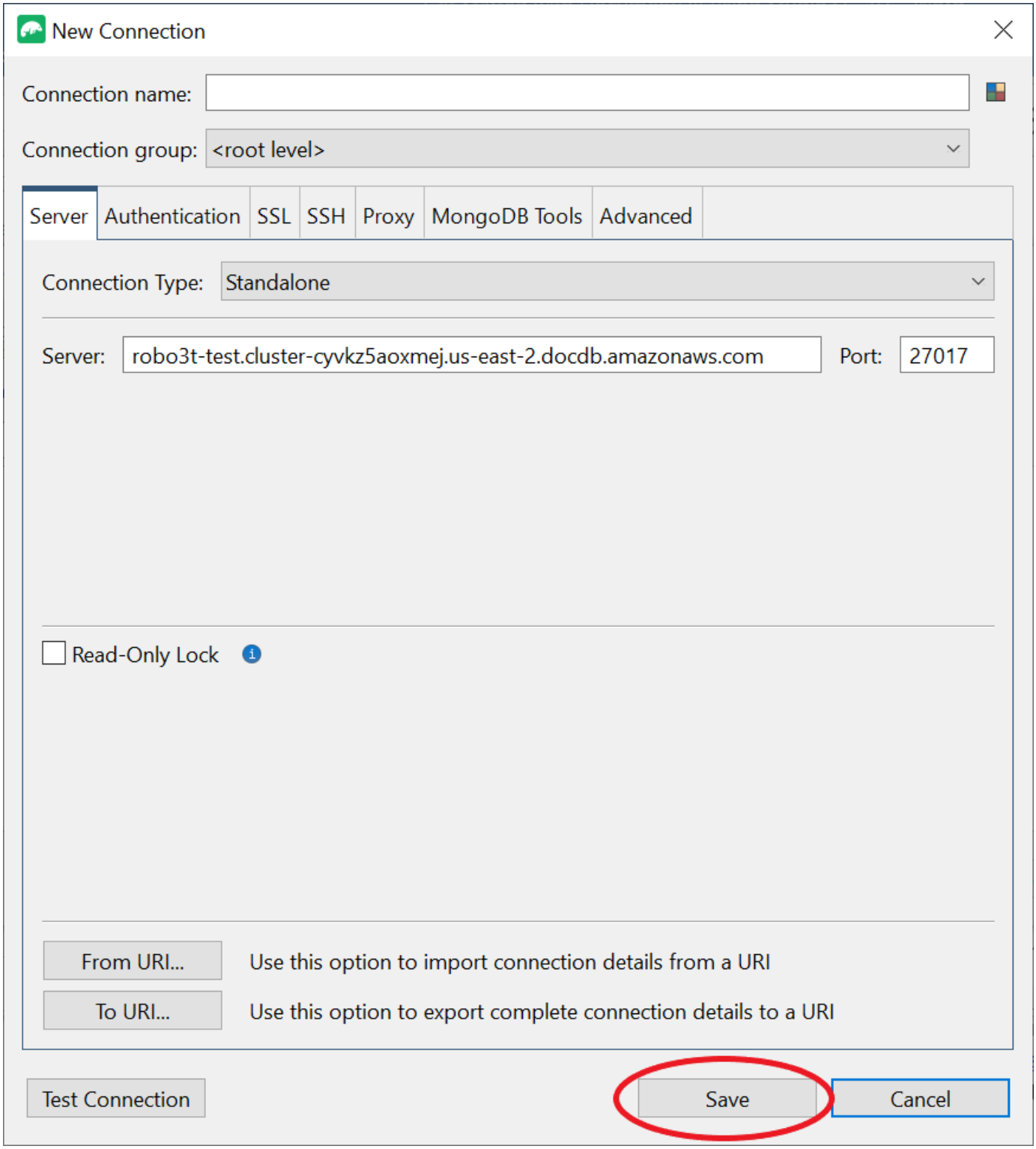
9. Testen Sie nun Ihre Konfiguration, indem Sie auf die Schaltfläche Verbindung testen klicken.



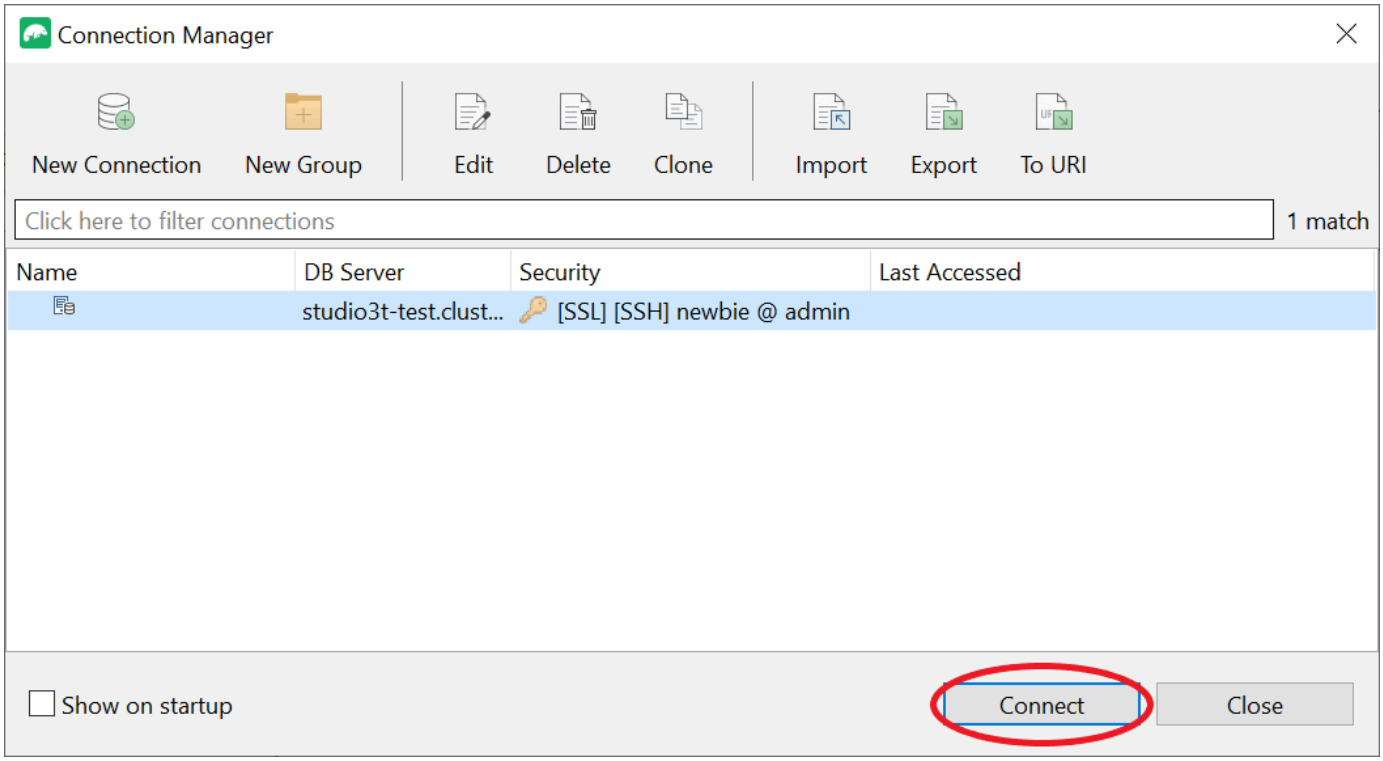
10. In einem Diagnosefenster sollte ein grüner Balken angezeigt werden, der anzeigt, dass der Test erfolgreich war. Wählen Sie nun OK, um das Diagnosefenster zu schließen.



11. Wählen Sie Speichern, um Ihre Verbindung für die future Verwendung zu speichern.



12. Wählen Sie nun Ihren Cluster aus und wählen Sie Connect.



Herzlichen Glückwunsch! Sie sind jetzt erfolgreich über Studio 3T mit Ihrem Amazon DocumentDB-Cluster verbunden.

Connect zu Amazon DocumentDB her mit DataGrip

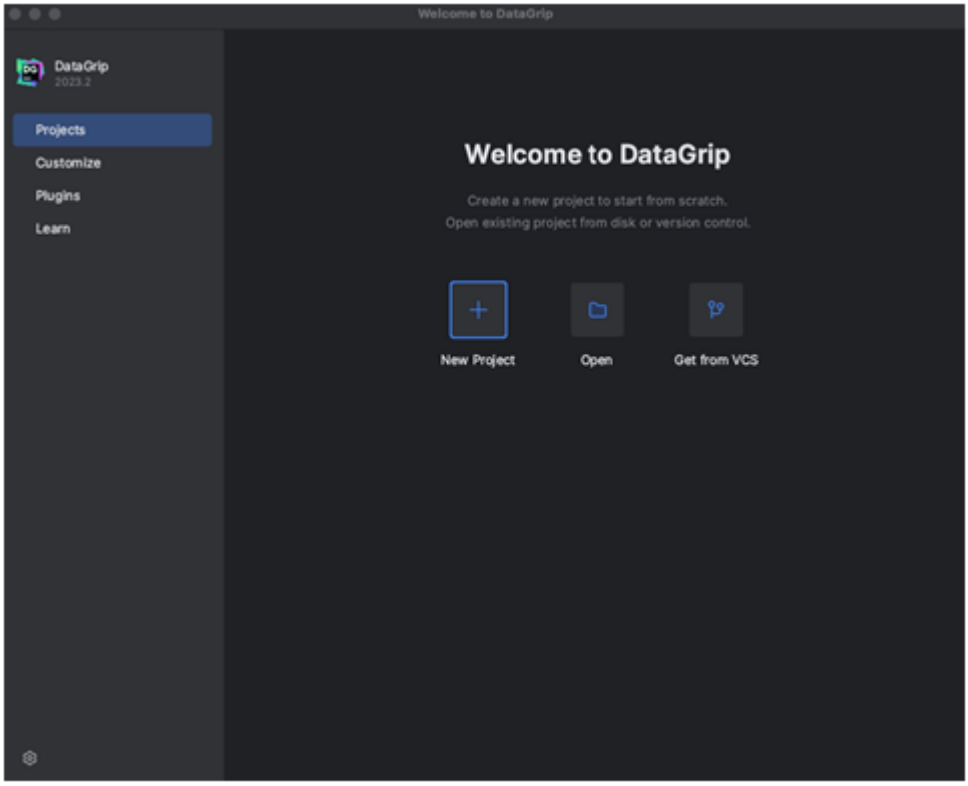
DataGripist eine leistungsstarke integrierte Entwicklungsumgebung (IDE), die verschiedene Datenbanksysteme unterstützt, darunter Amazon DocumentDB. Dieser Abschnitt führt Sie durch die Schritte, mit denen Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen DataGrip, sodass Sie Ihre Daten einfach über eine grafische Oberfläche verwalten und abfragen können.

Voraussetzungen

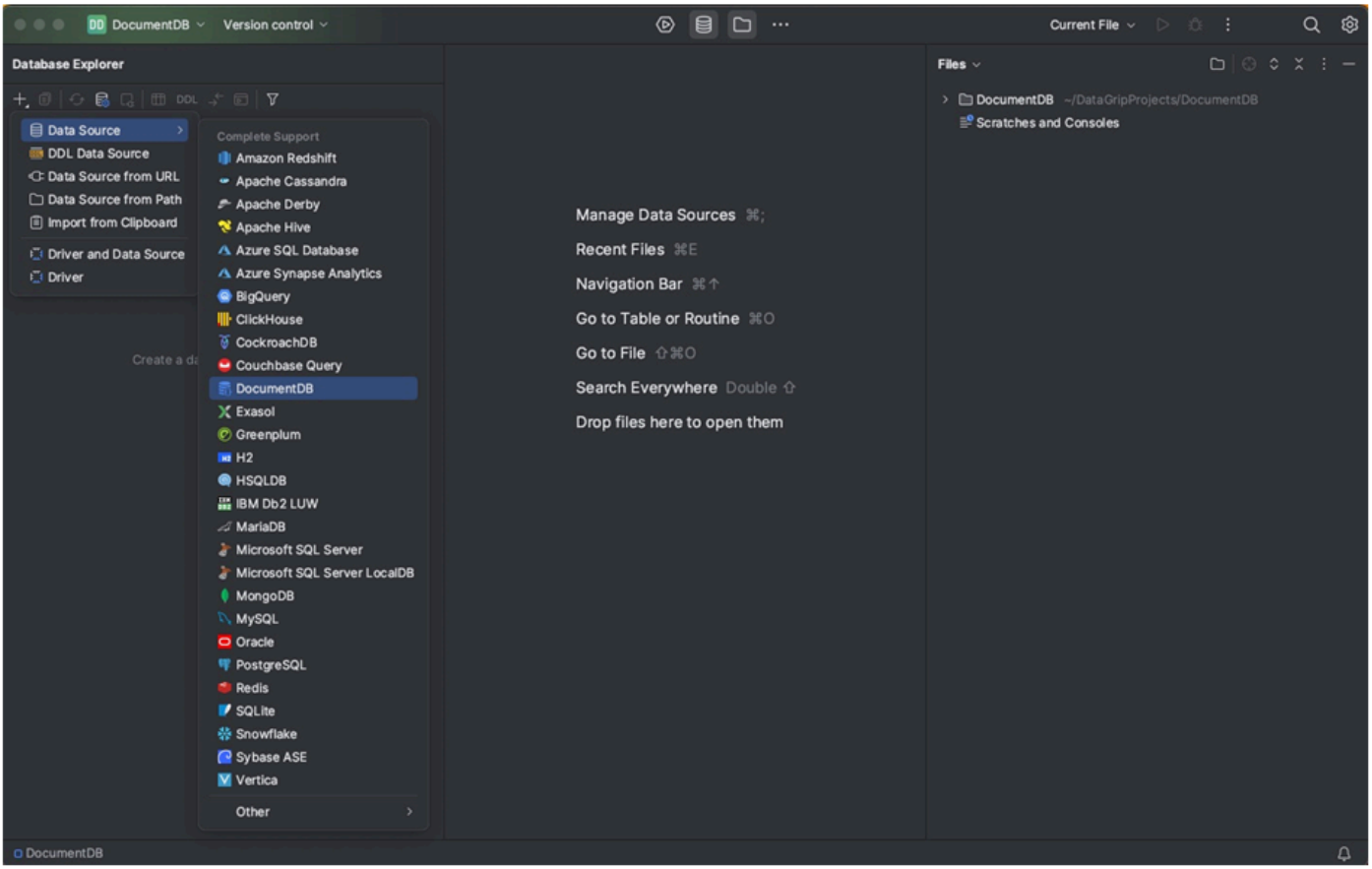
- DataGrip IDE ist auf Ihrem Computer installiert. Sie können es von herunterladen JetBrains.
- Eine EC2 Amazon-Instance, die in derselben VPC wie Ihr Amazon DocumentDB-Cluster läuft. Sie verwenden diese Instanz, um einen sicheren Tunnel von Ihrem lokalen Computer zum Amazon-Dokument einzurichtenDBcluster. Folgen Sie den Anweisungen dazuStellen Sie über Amazon eine Verbindung her EC2.
- Alternative zu einer EC2 Amazon-Instance, einer VPN-Verbindung oder wenn Sie bereits über ein sicheres VPN auf Ihre AWS Infrastruktur zugreifen. Wenn Sie diese Option bevorzugen, folgen Sie den Anweisungen für den sicheren Zugriff auf Amazon DocumentDB mit AWS Client VPN.

Connect mit DataGrip

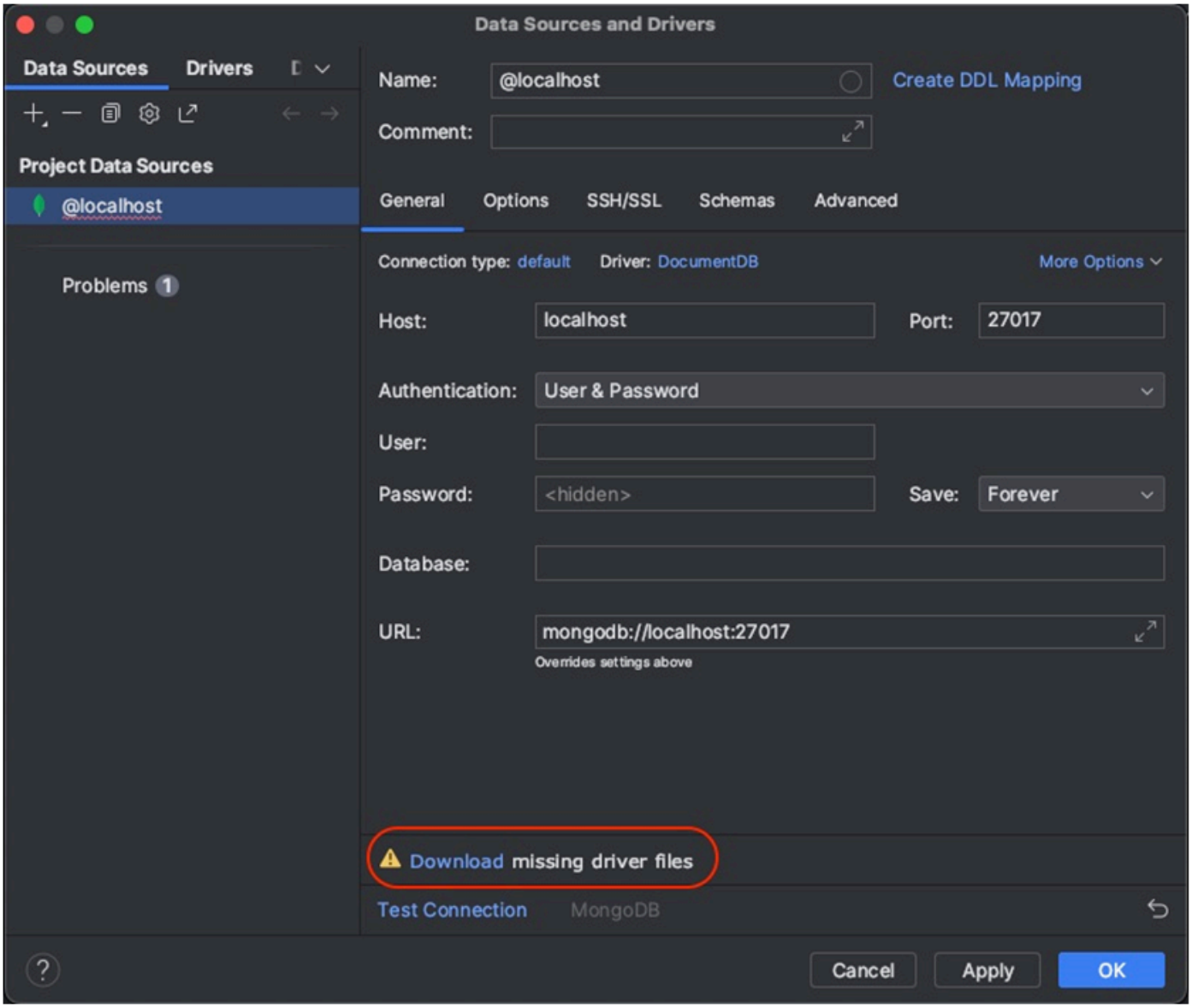
1. Starten Sie DataGrip auf Ihrem Computer und erstellen Sie ein neues Projekt.



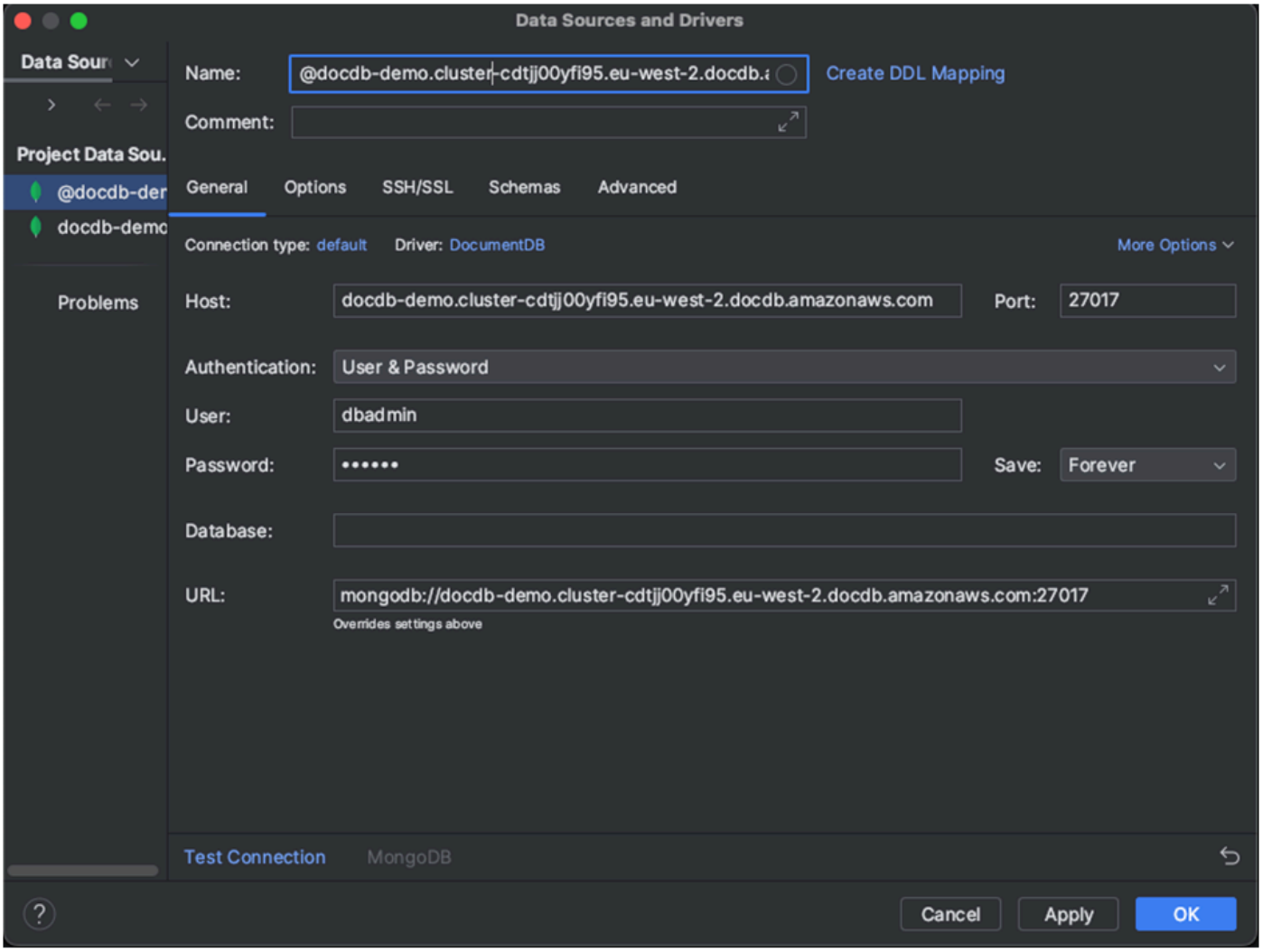
2. Fügen Sie mithilfe einer der folgenden Methoden eine neue Datenquelle hinzu:
   a. Navigieren Sie im Hauptmenü zu Datei — Neu — Datenquelle und wählen Sie DocumentDB
   b. Klicken Sie im Datenbank-Explorer in der Werkzeugleiste auf das neue Symbol (+). Navigieren Sie zu Data Source und wählen Sie DocumentDB aus.



3. Überprüfen Sie auf der Seite Datenquellen auf der Registerkarte Allgemein, ob sich unten im Bereich mit den Verbindungseinstellungen der Link Fehlende Treiberdateien herunterladen befindet. Klicken Sie auf diesen Link, um Treiber herunterzuladen, die für die Interaktion mit einer Datenbank erforderlich sind. Einen direkten Download-Link finden Sie unter JetBrains JDBC-Treiber.



4. Geben Sie auf der Registerkarte Allgemein die Verbindungsdetails an:
   a. Geben Sie im Feld Host den Amazon DocumentDB-Cluster-Endpunkt an.
   b. Der Port ist bereits auf 27017 eingestellt. Ändern Sie es, wenn Ihr Cluster auf einem anderen Port bereitgestellt wurde.
   c. Wählen Sie für Authentifizierung die Option Benutzer und Passwort aus.
   d. Geben Sie Ihren Benutzernamen und Ihr Passwort ein.
   e. Das Datenbankfeld ist optional. Sie können die Datenbank angeben, zu der Sie eine Verbindung herstellen möchten.
   f. Das URL-Feld wird automatisch vervollständigt, wenn Sie die obigen Details hinzufügen.



5. Aktivieren Sie auf der Registerkarte SSH/SSL die Option SSH-Tunnel verwenden und klicken Sie dann auf das Symbol, um den SSH-Konfigurationsdialog zu öffnen. Geben Sie die folgenden Informationen ein:

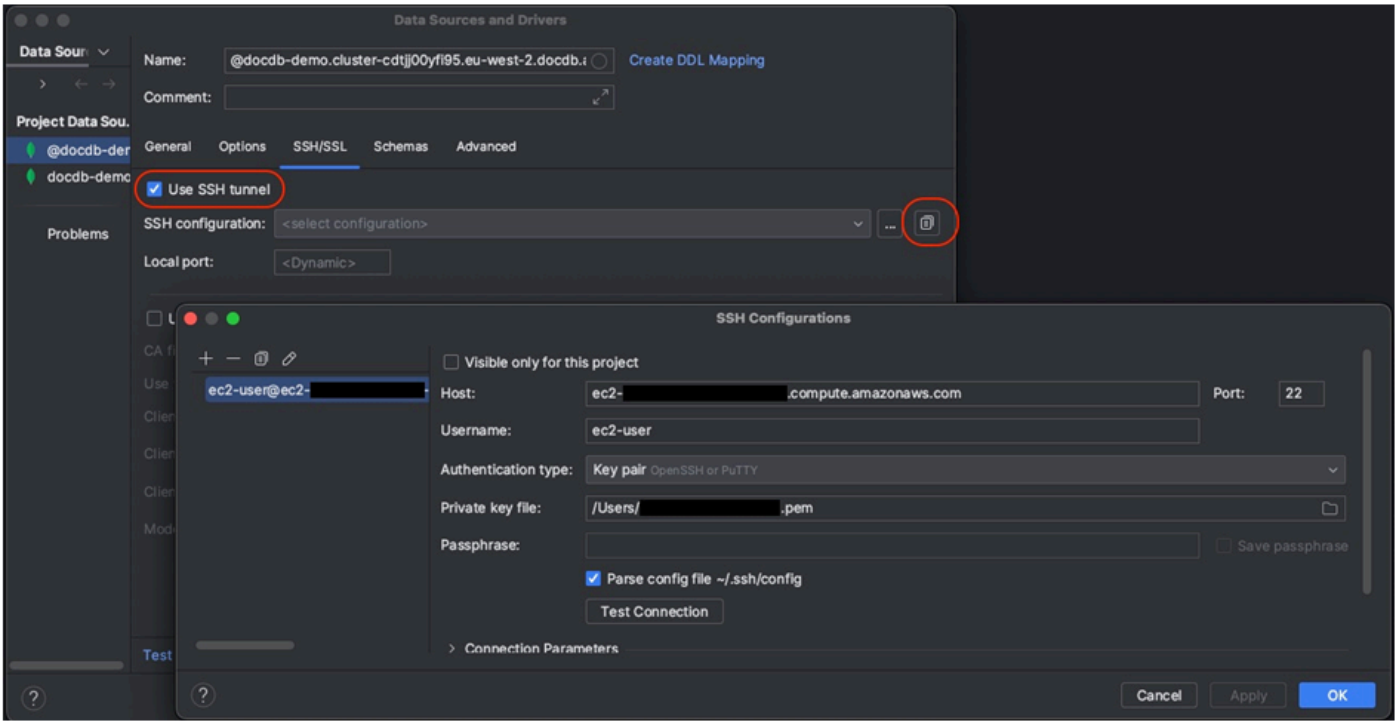
   a. Geben Sie im Feld Host den Hostnamen Ihrer EC2 Amazon-Instance ein.

   b. Geben Sie den Benutzernamen und das Passwort für Ihre EC2 Amazon-Instance ein.

   c. Wählen Sie als Authentifizierungstyp die Option Schlüsselpaar aus.

   d. Geben Sie Ihre private Schlüsseldatei ein.

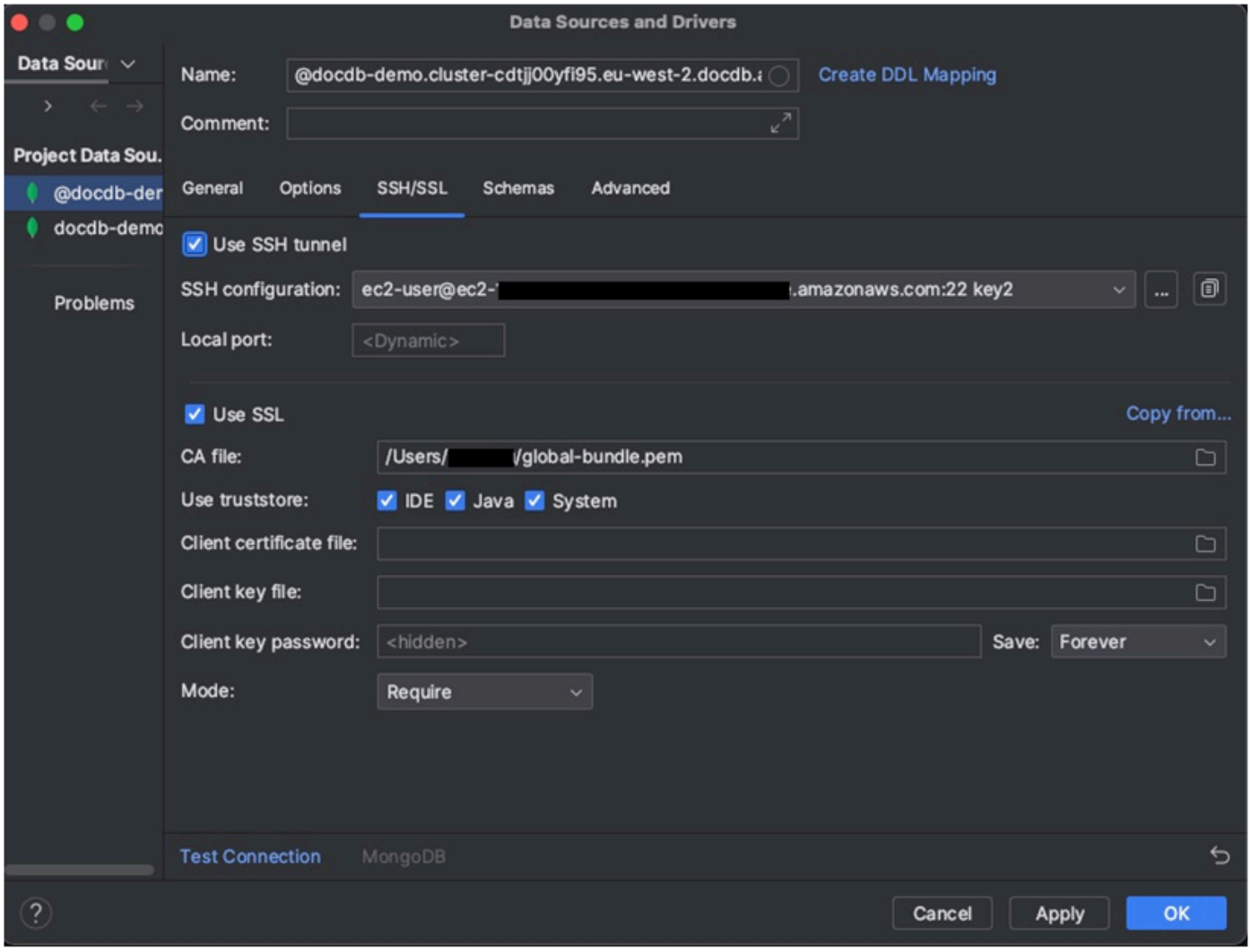
> **Note**
>
> Wenn Sie die VPN-Option verwenden, müssen Sie den SSH-Tunnel nicht konfigurieren.



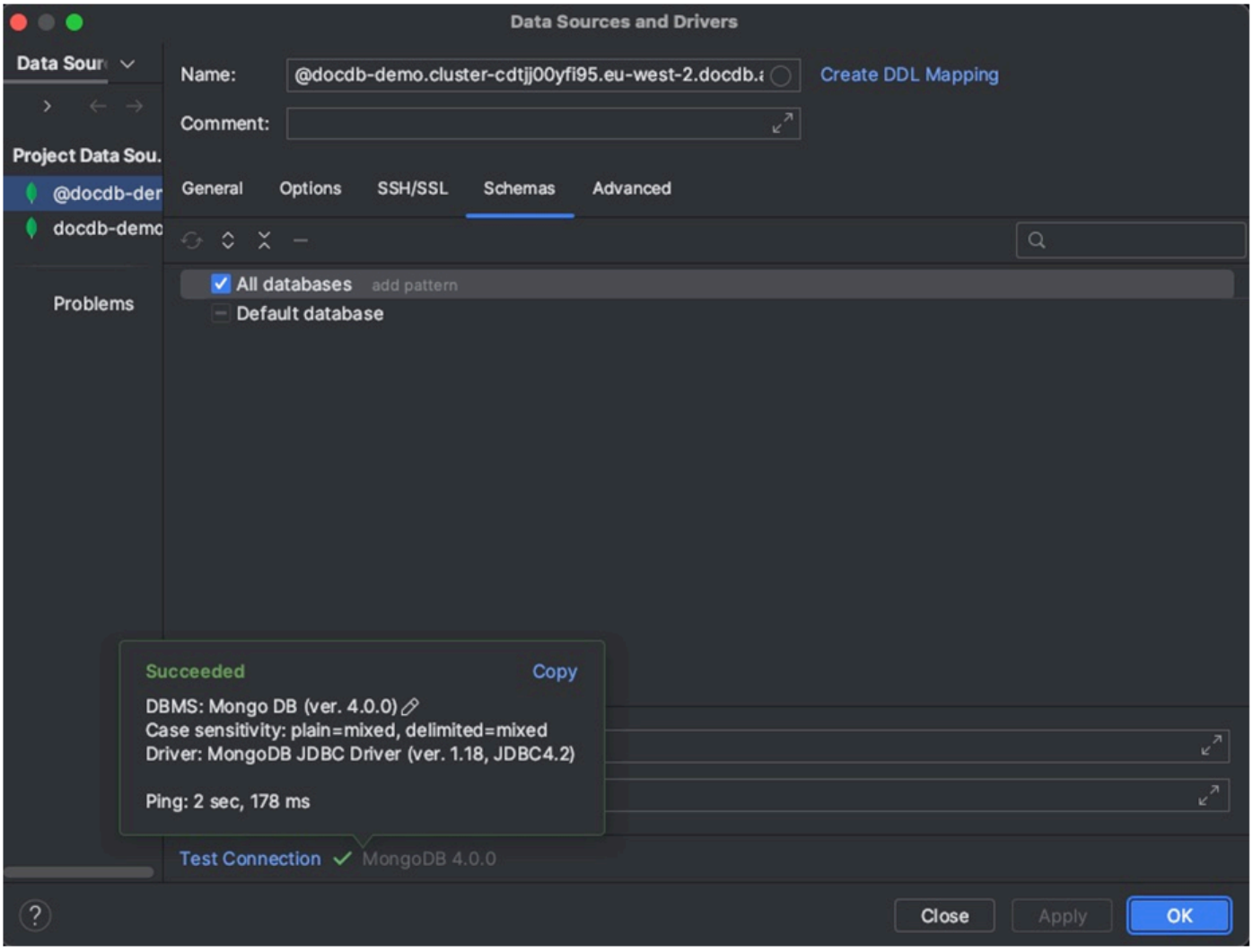
6. Aktivieren Sie auf der Registerkarte SSH/SSL die Option SSL verwenden. Geben Sie im Feld CA-Datei den Speicherort der `global-bundle.pem` Datei auf Ihrem Computer ein. Behalten Sie für Modus die Option Erforderlich bei.

> **Note**
>
> Sie können das Zertifikat von diesem Speicherort oder mit dem folgenden Befehl herunterladen: wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem

> **Note**
>
> Wenn Sie eine Verbindung zu Amazon DocumentDB Elastic Cluster herstellen, müssen Sie die CA-Datei nicht angeben. Lassen Sie die Option SSL verwenden aktiviert und lassen Sie alle anderen Optionen auf ihren Standardwerten stehen.



7. Wählen Sie auf der Registerkarte Schemas die Option Alle Datenbanken aus, oder geben Sie den Filter „*: *“ in das Feld Schemamuster ein. Klicken Sie auf den Link Verbindung testen, um die Verbindung zu testen.



8. Sobald die Verbindung erfolgreich getestet wurde, klicken Sie auf OK, um die Datenquellenkonfiguration zu speichern.

DataGrip features

DataGrip bietet verschiedene Funktionen, die Ihnen helfen, effizient mit Amazon DocumentDB zu arbeiten:

- SQL Editor — Schreiben Sie SQL-ähnliche Abfragen in Ihren DocumentDB-Sammlungen und führen Sie sie mit dem SQL-Editor in aus. DataGrip
- Visual Query Builder — Verwenden Sie den Visual Query Builder, um Abfragen grafisch zu erstellen, ohne SQL-Code schreiben zu müssen.
- Schemaverwaltung — Einfache Verwaltung Ihres Datenbankschemas, einschließlich des Erstellens, Änderns und Löschens von Sammlungen.

- Datenvisualisierung — Zeigen Sie Ihre Daten an und analysieren Sie sie mithilfe verschiedener Visualisierungstools, die unter verfügbar sind. DataGrip
- Daten exportieren und importieren — Übertragen Sie Daten zwischen Amazon DocumentDB und anderen Datenbanken mithilfe DataGrip der Export- und Importfunktionen.

Weiterführende Funktionen und Tipps zur Arbeit mit Amazon DocumentDB und anderen Datenbanksystemen finden Sie in der offiziellen DataGrip Dokumentation.

Stellen Sie über Amazon eine Verbindung her EC2

In diesem Abschnitt wird beschrieben, wie Sie die Konnektivität zwischen einem Amazon DocumentDB-Cluster und Amazon einrichten und von der EC2 Amazon-Instance aus auf den Amazon DocumentDB-Cluster zugreifen. EC2

Es gibt zwei Optionen für die Konfiguration der EC2 Verbindung:

- Automatisch Ihre EC2 Instance mit einer Amazon DocumentDB DocumentDB-Datenbank verbinden — Verwenden Sie die automatische Verbindungsfunktion in der EC2 Konsole, um die Verbindung zwischen Ihrer EC2 Instance und einer neuen oder vorhandenen Amazon DocumentDB DocumentDB-Datenbank automatisch zu konfigurieren. Diese Verbindung ermöglicht den Verkehr zwischen der EC2 Instance und der Amazon DocumentDB DocumentDB-Datenbank. Diese Option wird in der Regel zum Testen und Erstellen neuer Sicherheitsgruppen verwendet.
- Manuelles Verbinden Ihrer EC2 Instance mit Ihrer Amazon DocumentDB DocumentDB-Datenbank — Konfigurieren Sie die Verbindung zwischen Ihrer EC2 Instance und Ihrer Amazon DocumentDB DocumentDB-Datenbank, indem Sie die Sicherheitsgruppen manuell konfigurieren und zuweisen, um die Konfiguration zu reproduzieren, die durch die automatische Verbindungsfunktion erstellt wird. Diese Option wird normalerweise verwendet, um erweiterte Einstellungen zu ändern und bestehende Sicherheitsgruppen zu verwenden.

Voraussetzungen

Unabhängig von der Option und bevor Sie Ihren ersten Amazon DocumentDB-Cluster erstellen, müssen Sie Folgendes tun:

Erstellen Sie ein Amazon Web Services (AWS) -Konto

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

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

Um sich für eine anzumelden AWS-Konto

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

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen einen Verifizierungscode über die Telefontasten eingeben.

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

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

Für den Zugriff auf die Verwaltung von Amazon DocumentDB DocumentDB-Ressourcen wie Clustern, Instances und Cluster-Parametergruppen sind Anmeldeinformationen erforderlich, mit denen Sie Ihre Anfragen authentifizieren AWS können. Weitere Informationen finden Sie unter Identity and Access Management für Amazon DocumentDB.

1. Geben Sie in der Suchleiste von IAM ein und wählen Sie im angezeigten Drop-down-Menü IAM aus. AWS Management Console
2. Sobald Sie sich in der IAM-Konsole befinden, wählen Sie im Navigationsbereich Benutzer aus.
3. Wählen Sie Ihren Nutzernamen aus.
4. Klicken Sie auf den Button Berechtigungen hinzufügen.

5. Wählen Sie die Option Attach existing policies directly (Vorhandene Richtlinien direkt anfügen) aus.
6. Geben Sie `AmazonDocDBFullAccess` in die Suchleiste ein und wählen Sie sie aus, sobald sie in den Suchergebnissen erscheint.
7. Klicken Sie unten auf die blaue Schaltfläche mit der Aufschrift Weiter: Überprüfen.
8. Klicken Sie unten auf die blaue Schaltfläche mit der Aufschrift Berechtigungen hinzufügen.

Erstellen Sie eine Amazon Virtual Private Cloud (Amazon VPC)

Je nachdem, in welchem Bereich AWS-Region Sie sich befinden, haben Sie möglicherweise bereits eine Standard-VPC erstellt oder auch nicht. Wenn Sie keine Standard-VPC haben, führen Sie Schritt 1 der Anleitung Erste Schritte mit Amazon VPC im Amazon VPC-Benutzerhandbuch durch. Dies dauert weniger als fünf Minuten.

Amazon EC2 automatisch Connect

Themen

- Automatisches Verbinden einer EC2 Instance mit einer neuen Amazon DocumentDB DocumentDB-Datenbank
- Automatisches Verbinden einer EC2 Instance mit einer vorhandenen Amazon DocumentDB DocumentDB-Datenbank
- Überblick über die automatische Konnektivität mit einer Instance EC2
- Anzeigen verbundener Rechenressourcen

Bevor Sie eine Verbindung zwischen einer EC2 Instance und einer neuen Amazon DocumentDB DocumentDB-Datenbank einrichten, stellen Sie sicher, dass Sie die unter beschriebenen Anforderungen erfüllen. Überblick über die automatische Konnektivität mit einer Instance EC2 Wenn Sie nach der Konfiguration der Konnektivität Änderungen an Sicherheitsgruppen vornehmen, können sich die Änderungen auf die Verbindung zwischen der EC2 Instance und der Amazon DocumentDB DocumentDB-Datenbank auswirken.

> **Note**
>
> Sie können nur automatisch eine Verbindung zwischen einer EC2 Instance und einer Amazon DocumentDB DocumentDB-Datenbank einrichten, indem Sie die AWS Management Console

verwenden. Sie können keine automatische Verbindung mit der AWS CLI oder der Amazon DocumentDB DocumentDB-API einrichten.

Automatisches Verbinden einer EC2 Instance mit einer neuen Amazon DocumentDB DocumentDB-Datenbank

Beim folgenden Prozess wird davon ausgegangen, dass Sie die Schritte im Voraussetzungen Thema abgeschlossen haben.

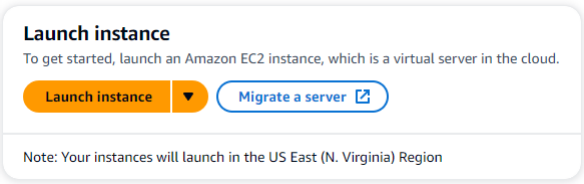
Schritte

- Schritt 1: EC2 Amazon-Instance erstellen
- Schritt 2: Erstellen Sie einen Amazon DocumentDB-Cluster
- Schritt 3: Connect zu Ihrer EC2 Amazon-Instance her
- Schritt 4: Installieren Sie die MongoDB-Shell
- Schritt 5: Amazon DocumentDB TLS verwalten
- Schritt 6: Connect zu Ihrem Amazon DocumentDB-Cluster her
- Schritt 7: Daten einfügen und abfragen
- Schritt 8: Erkunden

Schritt 1: EC2 Amazon-Instance erstellen

In diesem Schritt erstellen Sie eine EC2 Amazon-Instance in derselben Region und Amazon VPC, die Sie später für die Bereitstellung Ihres Amazon DocumentDB-Clusters verwenden werden.

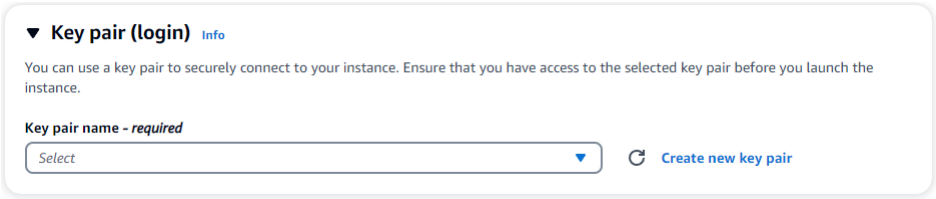
1. Wählen Sie auf der EC2 Amazon-Konsole Launch instance aus.



2. Geben Sie einen Namen oder eine Kennung in das Feld Name im Abschnitt Name und Tags ein.
3. Suchen Sie in der Dropdownliste Amazon Machine Image (AMI) nach Amazon Linux 2 AMI und wählen Sie es aus.



4. Suchen Sie in der Dropdownliste Instance-Typ nach t3.micro und wählen Sie es aus.
5. Geben Sie im Abschnitt key pair (Anmeldung) die Kennung eines vorhandenen Schlüsselpaars ein, oder wählen Sie Neues Schlüsselpaar erstellen.



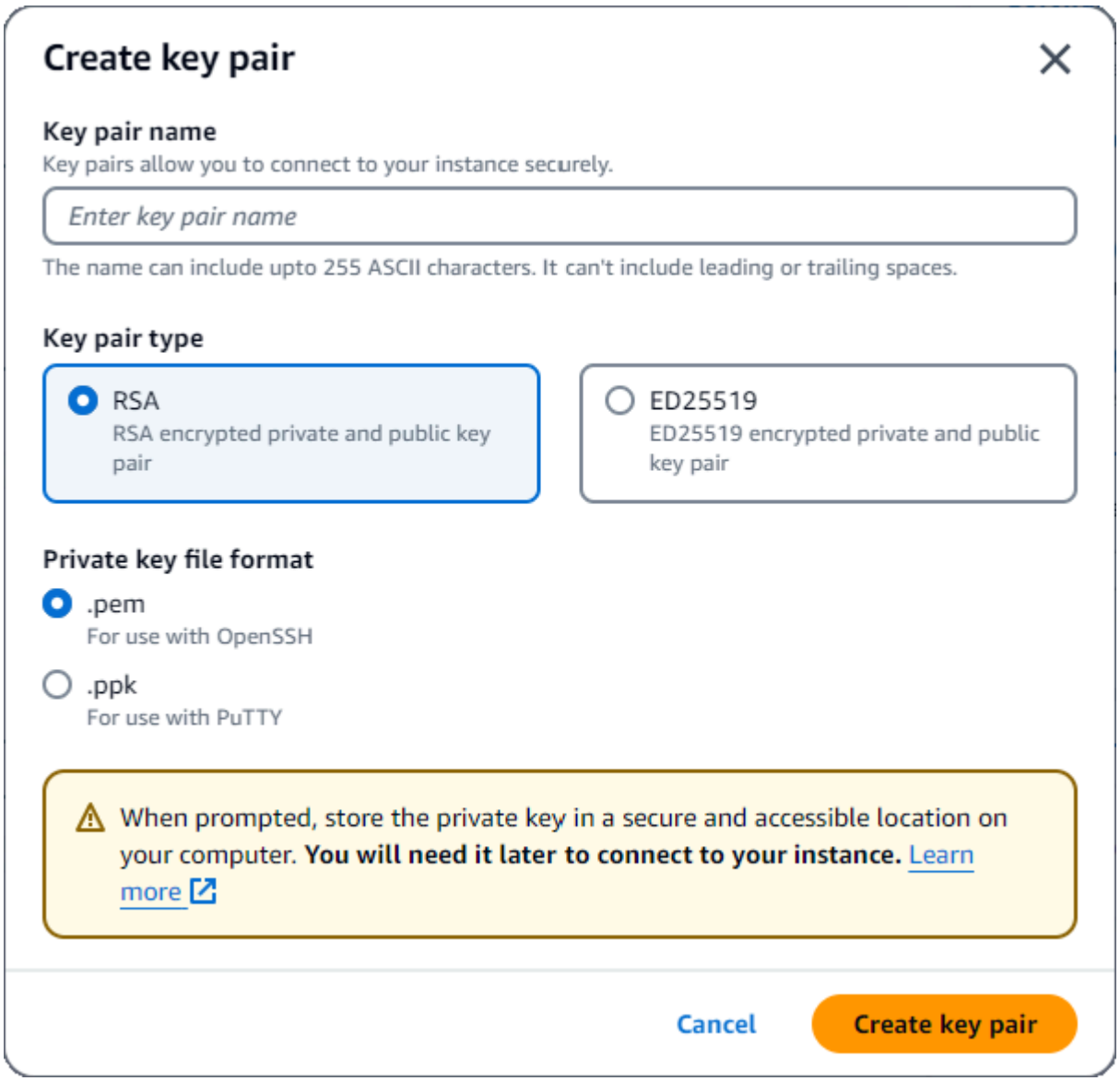
   Sie müssen ein EC2 Amazon-Schlüsselpaar angeben.

   Wenn Sie ein EC2 Amazon-Schlüsselpaar haben:

   a. Wählen Sie ein key pair aus, wählen Sie Ihr key pair aus der Liste.
   b. Sie müssen die private Schlüsseldatei (.pem- oder .ppk-Datei) bereits verfügbar haben, um sich bei Ihrer Amazon-Instance anmelden zu können. EC2

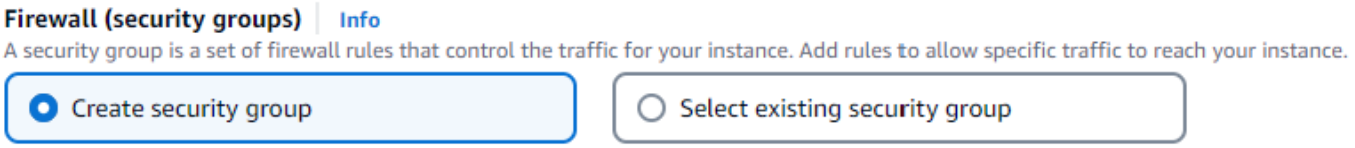
   Wenn Sie kein EC2 Amazon-Schlüsselpaar haben:

   a. Wählen Sie Neues key pair erstellen. Daraufhin wird das Dialogfeld key pair erstellen angezeigt.
   b. Geben Sie einen Namen in das Feld Name des Schlüsselpaars ein.
   c. Wählen Sie den Schlüsselpaartyp und das Dateiformat für den privaten Schlüssel.
   d. Wählen Sie Create Key Pair (Schlüsselpaar erstellen) aus.



> **Note**
>
> Aus Sicherheitsgründen empfehlen wir dringend, ein Schlüsselpaar sowohl für die SSH- als auch für die Internetverbindung zu Ihrer Instance zu verwenden. EC2

6. Optional: Wählen Sie im Bereich Netzwerkeinstellungen unter Firewall (Sicherheitsgruppen) die Option Sicherheitsgruppe erstellen aus.



Wählen Sie Sicherheitsgruppe erstellen (überprüfen Sie alle Regeln für die Zulassung von Datenverkehr, die für Ihre EC2 Konnektivität gelten).

> **Note**
>
> Wenn Sie eine bestehende Sicherheitsgruppe verwenden möchten, folgen Sie den Anweisungen unterAmazon EC2 manuell Connect.

7. Überprüfen Sie im Abschnitt Zusammenfassung Ihre EC2 Konfiguration und wählen Sie, falls korrekt, die Option Launch Instance aus.

Schritt 2: Erstellen Sie einen Amazon DocumentDB-Cluster

Während die EC2 Amazon-Instance bereitgestellt wird, erstellen Sie Ihren Amazon DocumentDB-Cluster.

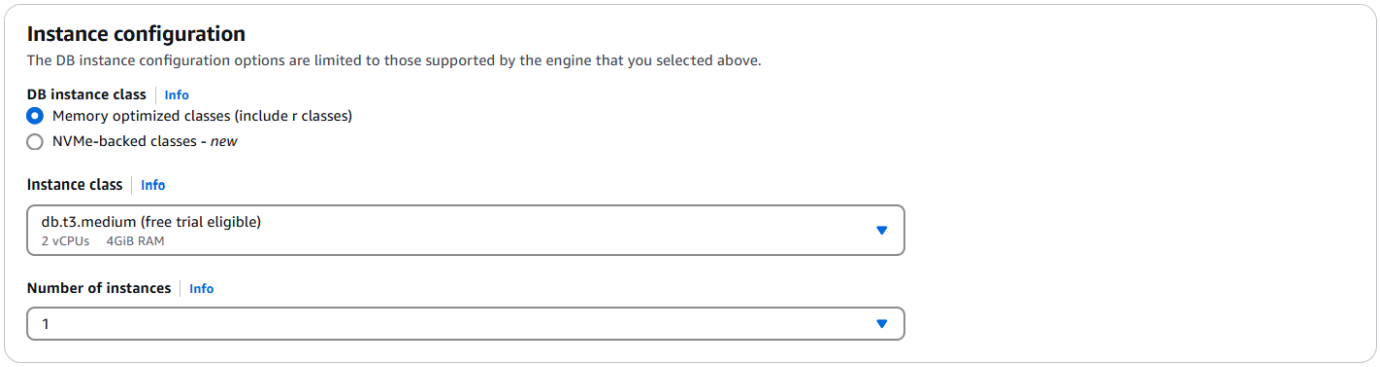
1. Navigieren Sie zur Amazon DocumentDB DocumentDB-Konsole und wählen Sie im Navigationsbereich Clusters aus.
2. Wählen Sie Erstellen aus.
3. Belassen Sie die Standardeinstellung für den Clustertyp auf Instance Based Cluster.
4. Geben Sie in Clusterkonfiguration für Cluster-ID einen eindeutigen Namen ein. Beachten Sie, dass die Konsole alle Clusternamen unabhängig davon, wie sie eingegeben wurden, in Kleinbuchstaben ändert.

   Behalten Sie für die Engine-Version den Standardwert 5.0.0 bei.
5. Behalten Sie für die Cluster-Speicherkonfiguration die Standardeinstellung von Amazon DocumentDB Standard bei.
6. In der Instanzkonfiguration:

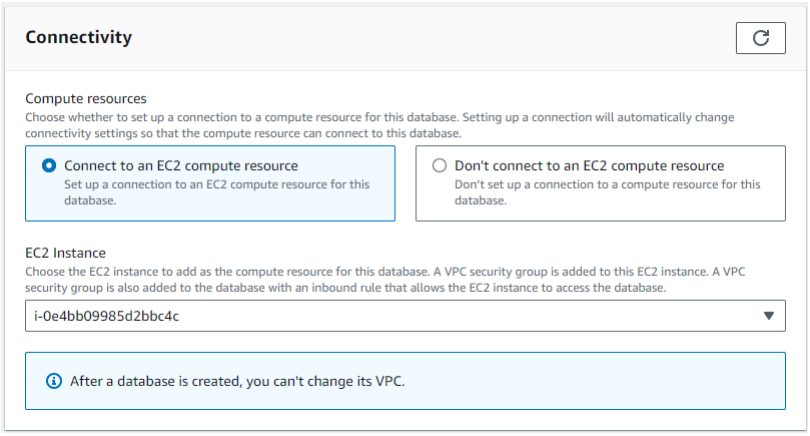
   - Wählen Sie für DB-Instance-Klasse die Option Speicheroptimierte Klassen (einschließlich R-Klassen) aus (dies ist die Standardeinstellung).

Bei der anderen Instance-Option handelt es sich um Klassen mit NVMe -backed. Weitere Informationen hierzu finden Sie unter NVMe-gestützte Instances.

- Wählen Sie für Instance-Klasse den Instance-Typ aus, der Ihren Anforderungen entspricht. Eine detailliertere Erläuterung der Instanzklassen finden Sie unterSpezifikationen der Instanzklasse.
- Wählen Sie für die Anzahl der Instances eine Zahl aus, die Ihren Anforderungen am besten entspricht. Denken Sie daran, je niedriger die Anzahl, desto niedriger die Kosten und desto geringer das read/write Volumen, das vom Cluster verwaltet werden kann.



7. Wählen Sie für Konnektivität die Option Mit einer EC2 Rechenressource Connect aus. Dies ist die EC2 Instanz, die Sie in Schritt 1 erstellt haben.



> **Note**
>
> Wenn Sie eine Verbindung zu einer EC2 Rechenressource herstellen, wird automatisch eine Sicherheitsgruppe für Ihre EC2 Rechenressourcenverbindung zu Ihrem Amazon DocumentDB-Cluster erstellt. Wenn Sie mit der Erstellung Ihres Clusters fertig sind und die neu erstellte Sicherheitsgruppe sehen

möchten, navigieren Sie zur Cluster-Liste und wählen Sie die Kennung Ihres Clusters aus. Gehen Sie auf der Registerkarte Konnektivität und Sicherheit zu Sicherheitsgruppen und suchen Sie unter Sicherheitsgruppenname (ID) nach Ihrer Gruppe. Sie wird den Namen Ihres Clusters enthalten und etwa wie folgt aussehen:`docdb-ec2-docdb-2023-12-11-21-33-41:i-0e4bb09985d2bbc4c (sg-0238e0b0bf0f73877)`.

8. Geben Sie im Abschnitt Authentifizierung einen Benutzernamen für den Hauptbenutzer ein und wählen Sie dann Selbstverwaltet aus. Geben Sie ein Passwort ein und bestätigen Sie es.

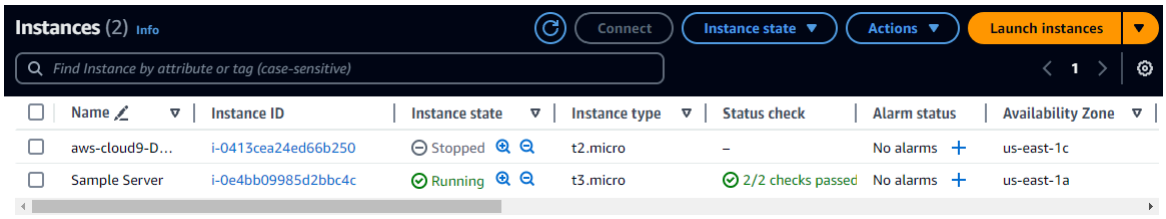
   Wenn Sie stattdessen Verwaltet in wählen AWS Secrets Manager, finden Sie Passwortverwaltung mit Amazon DocumentDB und AWS Secrets Manager weitere Informationen unter.

9. Wählen Sie Cluster erstellen.

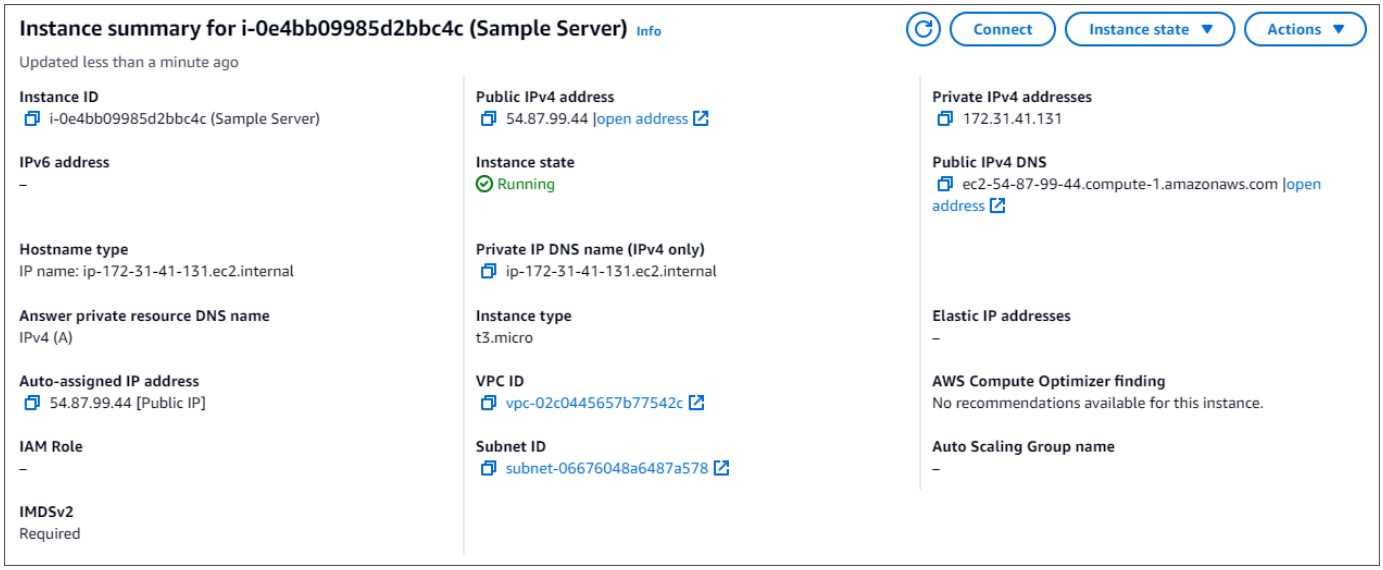
Schritt 3: Connect zu Ihrer EC2 Amazon-Instance her

Um die Mongo-Shell zu installieren, müssen Sie zunächst eine Verbindung zu Ihrer EC2 Amazon-Instance herstellen. Durch die Installation der Mongo-Shell können Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen und ihn abfragen. Führen Sie folgende Schritte aus:

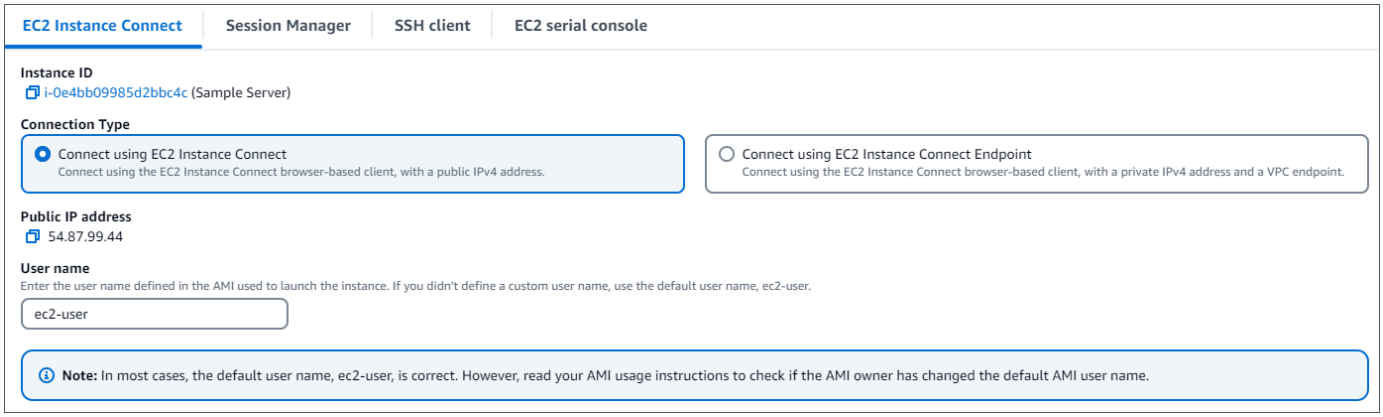
1. Navigieren Sie auf der EC2 Amazon-Konsole zu Ihren Instances und überprüfen Sie, ob die gerade erstellte Instance läuft. Wenn ja, wählen Sie die Instance aus, indem Sie auf die Instance-ID klicken.



2. Wählen Sie Connect aus.



3. Es gibt vier Optionen mit Registerkarten für Ihre Verbindungsmethode: Amazon EC2 Instance Connect, Session Manager, SSH-Client oder EC2 serielle Konsole. Sie müssen eine auswählen und deren Anweisungen befolgen. Wenn Sie fertig sind, wählen Sie Connect.



> **Note**
>
> Wenn sich Ihre IP-Adresse geändert hat, nachdem Sie diesen Walkthrough gestartet haben, oder wenn Sie zu einem späteren Zeitpunkt zu Ihrer Umgebung zurückkehren, müssen Sie Ihre demoEC2 Sicherheitsgruppenregel für eingehenden Datenverkehr aktualisieren, um eingehenden Datenverkehr von Ihrer neuen API-Adresse aus zu ermöglichen.

Schritt 4: Installieren Sie die MongoDB-Shell

Sie können jetzt die MongoDB-Shell installieren. Dabei handelt es sich um ein Befehlszeilenprogramm, mit dem Sie eine Verbindung herstellen und Ihren Amazon DocumentDB-Cluster abfragen. Derzeit gibt es zwei Versionen der MongoDB-Shell: die neueste Version, Mongosh, und die vorherige Version, Mongo-Shell.

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Folgen Sie den nachstehenden Anweisungen, um die MongoDB-Shell für Ihr Betriebssystem zu installieren.

On Amazon Linux

So installieren Sie die MongoDB-Shell auf Amazon Linux

Wenn Sie keine IAM-Authentifizierung verwenden und die neueste MongoDB-Shell (mongosh) verwenden möchten, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, gehen Sie wie folgt vor:

1. Erstellen Sie die Repository-Datei. Führen Sie in der Befehlszeile Ihrer EC2 Instanz, die Sie erstellt haben, den folgenden Befehl aus:

   ```
   echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
   repo.mongodb.org/yum/amazon/2023/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
    \ngpgkey=https://pgp.mongodb.com/server-5.0.asc" | sudo tee /etc/yum.repos.d/
   mongodb-org-5.0.repo
   ```

2. Wenn der Vorgang abgeschlossen ist, installieren Sie mongosh mit einer der beiden folgenden Befehlsoptionen an der Befehlszeile:

   Option 1 — Wenn Sie während der EC2 Amazon-Konfiguration das Standard-Amazon Linux 2023 ausgewählt haben, geben Sie diesen Befehl ein:

```
sudo yum install -y mongodb-mongosh-shared-openssl3
```

Option 2 — Wenn Sie bei der EC2 Amazon-Konfiguration Amazon Linux 2 ausgewählt haben, geben Sie diesen Befehl ein:

```
sudo yum install -y mongodb-mongosh
```

Wenn Sie die IAM-Authentifizierung verwenden, müssen Sie die vorherige Version der MongoDB-Shell (5.0) verwenden, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen. Gehen Sie wie folgt vor:

1. Erstellen Sie die Repository-Datei. Führen Sie in der Befehlszeile Ihrer EC2 Instanz, die Sie erstellt haben, den folgenden Befehl aus:

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2023/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
 \ngpgkey=https://pgp.mongodb.com/server-5.0.asc" | sudo tee /etc/yum.repos.d/
mongodb-org-5.0.repo
```

2. Wenn der Vorgang abgeschlossen ist, installieren Sie die Mongodb 5.0-Shell mit der folgenden Befehlsoption an der Befehlszeile:

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

On Ubuntu

Um mongosh auf Ubuntu zu installieren

1. Importieren Sie den öffentlichen Schlüssel, der von dem Paketverwaltungssystem verwendet wird.

```
curl -fsSL https://pgp.mongodb.com/server-5.0.asc | sudo gpg --dearmor -o /usr/
share/keyrings/mongodb-server-5.0.gpg
```

2. Erstellen Sie die Listendatei `mongodb-org-5.0.list` für MongoDB mit dem korrekten Befehl für Ihre Ubuntu-Version.

```
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-
server-5.0.gpg ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0
 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list
```

3. Importieren und aktualisieren Sie die lokale Paketdatenbank mit dem folgenden Befehl:

```
sudo apt-get update
```

4. Installieren Sie Mongosh.

```
sudo apt-get install -y mongodb-mongosh
```

Weitere Informationen zum Installieren von früheren Versionen von MongoDB auf Ihrem Ubuntu-System finden Sie unter Installieren von MongoDB Community Edition auf Ubuntu.

On other operating systems

Informationen zum Installieren der mongo-Shell auf anderen Betriebssystemen finden Sie unter Installieren von MongoDB Community Edition in der MongoDB-Dokumentation.

Schritt 5: Amazon DocumentDB TLS verwalten

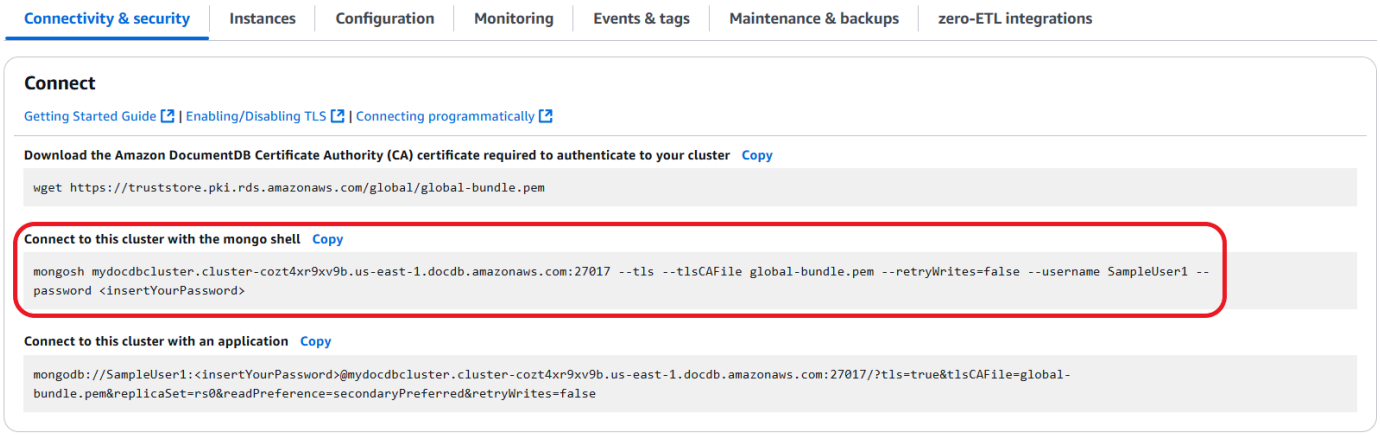
Laden Sie das CA-Zertifikat für Amazon DocumentDB mit dem folgenden Code herunter: `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

> **Note**
>
> Transport Layer Security (TLS) ist standardmäßig für alle neuen Amazon DocumentDB-Cluster aktiviert. Weitere Informationen finden Sie unter TLS-Einstellungen für Amazon DocumentDB-Cluster verwalten.

Schritt 6: Connect zu Ihrem Amazon DocumentDB-Cluster her

1. Suchen Sie auf der Amazon DocumentDB DocumentDB-Konsole unter Clusters Ihren Cluster. Wählen Sie den Cluster aus, den Sie erstellt haben, indem Sie auf die Cluster-ID für diesen Cluster klicken.

2. Suchen Sie auf der Registerkarte Konnektivität und Sicherheit im Feld Connect nach Connect to this cluster with the mongo shell:



   Kopieren Sie die angegebene Verbindungszeichenfolge und fügen Sie sie in Ihr Terminal ein.

   Nehmen Sie die folgenden Änderungen daran vor:

   a. Stellen Sie sicher, dass Sie den richtigen Benutzernamen in der Zeichenfolge haben.

   b. Auslassen, `<insertYourPassword>` sodass Sie beim Herstellen der Verbindung von der Mongo-Shell zur Eingabe des Passworts aufgefordert werden.

   c. Optional: Wenn Sie die IAM-Authentifizierung oder die vorherige Version der MongoDB-Shell verwenden, ändern Sie Ihre Verbindungszeichenfolge wie folgt:

      ```
      mongo --ssl --host mydocdbcluster.cluster-cozt4xr9xv9b.us-east-1.docdb.amazonaws.com:27017 --sslCAFile global-bundle.pem --username SampleUser1 --password
      ```

      `mydocdbcluster.cluster-cozt4xr9xv9b.us-east-1`Ersetzen Sie durch dieselben Informationen aus Ihrem Cluster.

3. Drücken Sie die Eingabetaste in Ihrem Terminal. Sie werden nun zur Eingabe Ihres Passworts aufgefordert. Geben Sie Ihr Passwort ein.

4. Wenn Sie Ihr Passwort eingeben und die `rs0 [direct: primary] <env-name>>` Aufforderung sehen, sind Sie erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden.

Haben Sie Probleme beim Herstellen einer Verbindung? Weitere Informationen finden Sie unter Fehlerbehebung bei Amazon DocumentDB.

Schritt 7: Daten einfügen und abfragen

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

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

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

   Sie erhalten die folgende Ausgabe:

```
{
  acknowledged: true,
  insertedId: ObjectId('673657216bdf6258466b128c')
}
```

2. Sie können das Dokument lesen, das Sie mit dem `findOne()` Befehl geschrieben haben (weil er nur ein einziges Dokument zurückgibt). Geben Sie Folgendes ein:

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

   Sie erhalten die folgende Ausgabe:

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

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

```
db.profiles.insertMany([{ _id: 1, name: 'Matt', status: 'active', level: 12, score:
 202 },
     { _id: 2, name: 'Frank', status: 'inactive', level: 2, score: 9 },
     { _id: 3, name: 'Karen', status: 'active', level: 7, score: 87 },
     { _id: 4, name: 'Katie', status: 'active', level: 3, score: 27 }
])
```

   Sie erhalten die folgende Ausgabe:

```
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3, '3': 4 } }
```

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

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

   Sie erhalten eine Ausgabe, die den in Schritt 3 eingegebenen Daten entspricht.

5. Verwenden Sie mithilfe eines Filters eine Abfrage für ein einzelnes Dokument. Geben Sie Folgendes ein:

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

   Sie erhalten die folgende Ausgabe:

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

6. Versuchen wir nun, ein Profil zu finden und es mit dem `findAndModify` Befehl zu ändern. Wir geben dem Benutzer Matt zusätzliche 10 Punkte mit dem folgenden Code:

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

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

```
{
    [{_id : 1, name : 'Matt', status: 'active', level: 12, score: 202}]
```

7. Sie können mit der folgenden Abfrage überprüfen, ob sich sein Punktestand geändert hat:

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

   Sie erhalten die folgende Ausgabe:

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

Schritt 8: Erkunden

Herzlichen Glückwunsch! Sie haben die Schnellstartanleitung für Amazon DocumentDB erfolgreich abgeschlossen.

Was kommt als Nächstes? Erfahren Sie, wie Sie diese leistungsstarke Datenbank mit einigen ihrer beliebten Funktionen voll ausschöpfen können:

- Amazon DocumentDB verwalten
- Skalierung
- Sichern und Wiederherstellen

> **Note**
>
> Um Kosten zu sparen, können Sie entweder Ihren Amazon DocumentDB-Cluster stoppen, um die Kosten zu senken, oder den Cluster löschen. Standardmäßig stoppt Ihre AWS Cloud9 Umgebung nach 30 Minuten Inaktivität die zugrunde liegende EC2 Amazon-Instance.

Automatisches Verbinden einer EC2 Instance mit einer vorhandenen Amazon DocumentDB DocumentDB-Datenbank

Das folgende Verfahren setzt voraus, dass Sie über einen vorhandenen Amazon DocumentDB-Cluster und eine bestehende EC2 Amazon-Instance verfügen.

Greifen Sie auf Ihren Amazon DocumentDB-Cluster zu und richten Sie die Amazon-Verbindung EC2 ein

1. Greifen Sie auf Ihren Amazon DocumentDB-Cluster zu.

    a. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.

    b. Klicken Sie im Navigationsbereich auf Cluster.

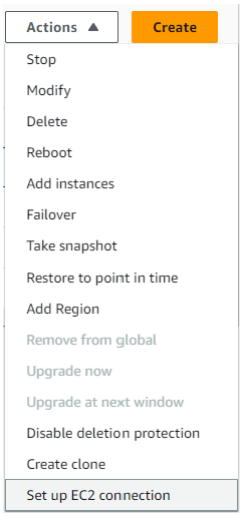
    > **Tip**
    >
    > Wenn der Navigationsbereich auf der linken Seite des Bildschirms nicht angezeigt wird, wählen Sie links oben auf der Seite das Menüsymbol

(≡)
aus.

c. Geben Sie den gewünschten Cluster an, indem Sie auf die Schaltfläche links neben dem Namen des Clusters klicken.

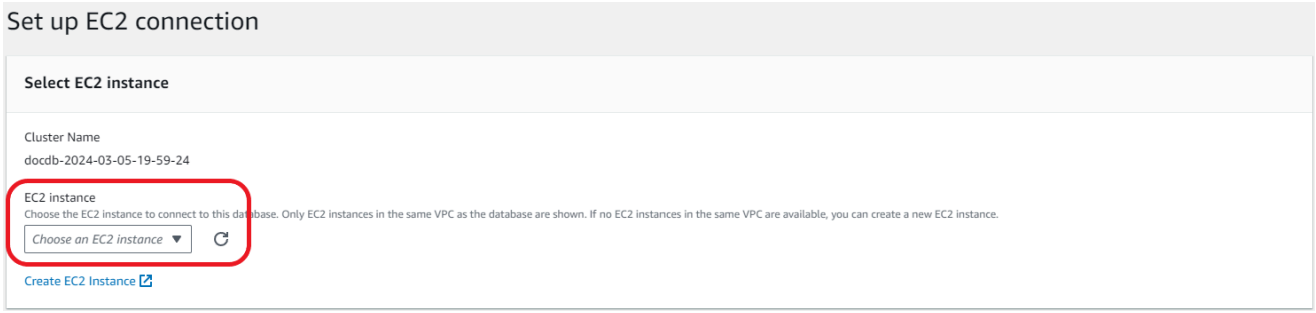
2. Richten Sie die EC2 Amazon-Verbindung ein.

a. Wählen Sie Aktionen und dann EC2 Verbindung einrichten.



Das Dialogfeld EC2 Verbindung einrichten wird angezeigt.

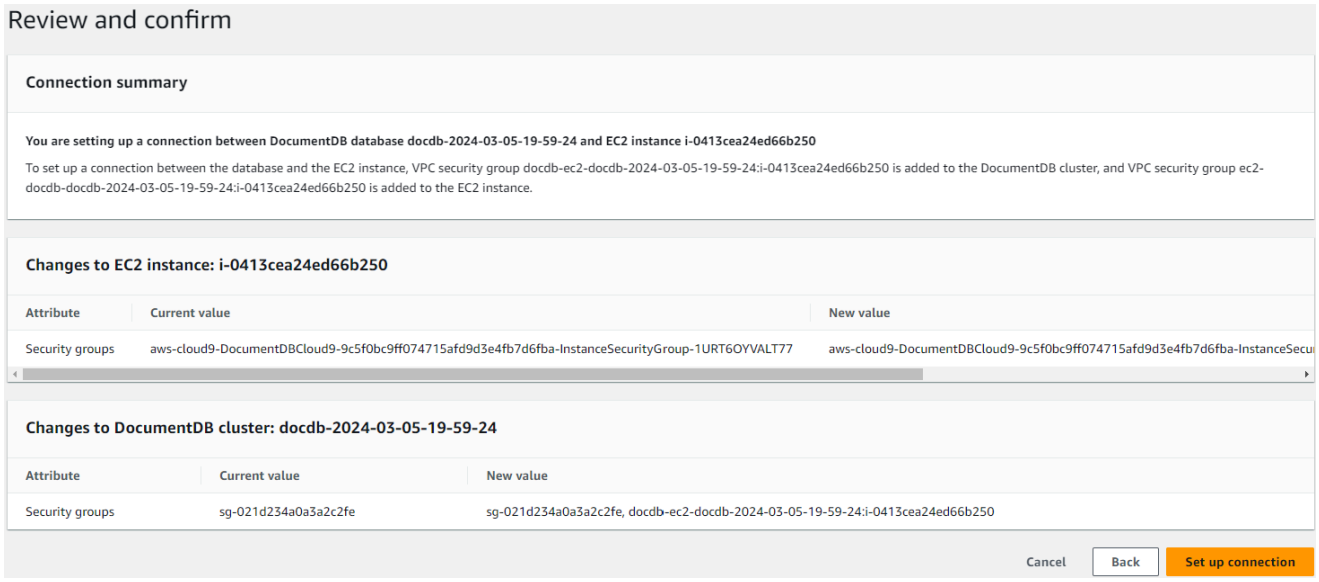
b. Wählen Sie im EC2 Instanzfeld die EC2 Instance aus, die Sie mit Ihrem Cluster verbinden möchten.



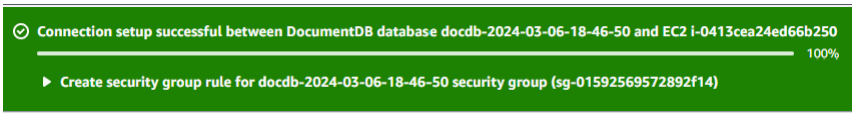
c. Klicken Sie auf Weiter.

Das Dialogfeld „Überprüfen und bestätigen“ wird angezeigt.

d. Vergewissern Sie sich, dass die Änderungen korrekt sind. Wählen Sie dann Verbindung einrichten.



Bei Erfolg wird die folgende Überprüfung angezeigt:



Überblick über die automatische Konnektivität mit einer Instance EC2

Wenn Sie eine Verbindung zwischen einer EC2 Instance und einer Amazon DocumentDB-Datenbank einrichten, konfiguriert Amazon DocumentDB automatisch die VPC-Sicherheitsgruppe für Ihre EC2 Instance und für Ihre Amazon DocumentDB DocumentDB-Datenbank.

Die folgenden Anforderungen gelten für die Verbindung einer EC2 Instance mit einer Amazon DocumentDB DocumentDB-Datenbank:

- Die EC2 Instance muss in derselben VPC wie die Amazon DocumentDB DocumentDB-Datenbank existieren.

  Wenn in derselben VPC keine EC2 Instanzen vorhanden sind, bietet die Konsole einen Link zum Erstellen einer Instanz.
- Der Benutzer, der die Konnektivität einrichtet, muss über Berechtigungen verfügen, um die folgenden EC2 Amazon-Operationen auszuführen:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`

- `ec2:DescribeInstances`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:RevokeSecurityGroupEgress`

Wenn sich die DB-Instance und die EC2 Instance in unterschiedlichen Availability Zones befinden, können für Ihr Konto Kosten für mehrere Availability Zones anfallen.

Wenn Sie eine Verbindung zu einer EC2 Instance einrichten, handelt Amazon DocumentDB gemäß der aktuellen Konfiguration der Sicherheitsgruppen, die der Amazon DocumentDB DocumentDB-Datenbank und EC2 -Instance zugeordnet sind, wie in der folgenden Tabelle beschrieben:

Aktuelle Amazon DocumentDB-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration EC2 der Sicherheitsgruppe	Amazon DocumentDB DocumentDB-Aktion
Der Amazon DocumentDB DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine eingehende Regel mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle.	Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht `DocumentDB-ec2-n` (wobei n eine Zahl ist). Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC-Sicherheitsgruppe der Amazon DocumentDB DocumentDB-Datenbank als Quelle.	Amazon DocumentDB ergreift keine Maßnahmen. Eine Verbindung zwischen der EC2 Instance und der Amazon DocumentDB DocumentDB-Datenbank wurde bereits automatisch konfiguriert. Da bereits eine Verbindung zwischen der EC2 Instance und der Amazon DocumentDB DocumentDB-Datenbank besteht, werden die Sicherheitsgruppen nicht geändert.
Es gilt eine der folgenden Bedingungen:	Es gilt eine der folgenden Bedingungen:	Amazon DocumentDB DocumentDB-Aktion: Neue Sicherheitsgruppen erstellen

Aktuelle Amazon DocumentDB-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration EC2 der Sicherheitsgruppe	Amazon DocumentDB DocumentDB-Aktion
• Der Amazon DocumentDB DocumentDB-Datenbank ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. • Der Amazon DocumentDB sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2 Instance verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle hat. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde. Beispiele für Änderungen sind das Hinzufügen einer Regel oder das Ändern des Ports einer vorhandenen Regel.	• Der EC2 Instance ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `ec2-DocumentDB-n` entspricht. • Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entsprichtec2-DocumentDB-n . Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Amazon DocumentDB DocumentDB-Datenbank verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine einzige ausgehende Regel mit der VPC-Sicherheitsgruppe der Amazon DocumentDB DocumentDB-Datenbank als Quelle hat. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	

Aktuelle Amazon DocumentDB-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration EC2 der Sicherheitsgruppe	Amazon DocumentDB DocumentDB-Aktion
Der Amazon DocumentDB DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine eingehende Regel mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle.	Der EC2 Instanz sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster entspricht. `ec2-DocumentDB-n` Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der Amazon DocumentDB DocumentDB-Datenbank verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine einzige ausgehende Regel mit der VPC-Sicherheitsgruppe der Amazon DocumentDB DocumentDB-Datenbank als Quelle hat. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Amazon DocumentDB DocumentDB-Aktion: Neue Sicherheitsgruppen erstellen

Aktuelle Amazon DocumentDB-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration EC2 der Sicherheitsgruppe	Amazon DocumentDB DocumentDB-Aktion
Der Amazon DocumentDB DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine eingehende Regel mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle.	Eine gültige EC2 Sicherheitsgruppe für die Verbindung ist vorhanden, aber sie ist der EC2 Instance nicht zugeordnet. Die Sicherheitsgruppe trägt einen Namen, der dem Muster `DocumentDB-ec2-n` entspricht. Sie wurde nicht geändert. Es gibt nur eine ausgehende Regel mit der VPC-Sicherheitsgruppe der Amazon DocumentDB DocumentDB-Datenbank als Quelle.	Amazon DocumentDB DocumentDB-Aktion: EC2 Sicherheitsgruppe zuordnen

Aktuelle Amazon DocumentDB-Sicherheitsgruppenkonfiguration	Aktuelle Konfiguration EC2 der Sicherheitsgruppe	Amazon DocumentDB DocumentDB-Aktion
Es gilt eine der folgenden Bedingungen: • Der Amazon DocumentDB DocumentDB-Datenbank ist keine Sicherheitsgruppe zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. • Der Amazon DocumentDB DocumentDB-Datenbank sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Amazon DocumentDB kann jedoch keine dieser Sicherheitsgruppen für die Verbindung mit der EC2 Instance verwenden. Amazon DocumentDB kann keine Sicherheitsgruppe verwenden, die keine Regel für eingehenden Datenverkehr mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle hat. Amazon DocumentDB kann auch keine Sicherheitsgruppe verwenden, die geändert wurde.	Der EC2 Instance sind eine oder mehrere Sicherheitsgruppen zugeordnet, deren Name dem Muster `DocumentDB-ec2-n` entspricht. Eine Sicherheitsgruppe, die dem Muster entspricht, wurde nicht geändert. Diese Sicherheitsgruppe hat nur eine ausgehende Regel mit der VPC-Sicherheitsgruppe der Amazon DocumentDB DocumentDB-Datenbank als Quelle.	Amazon DocumentDB DocumentDB-Aktion: Neue Sicherheitsgruppen erstellen

Amazon DocumentDB DocumentDB-Aktion: Neue Sicherheitsgruppen erstellen

Amazon DocumentDB ergreift die folgenden Aktionen:

- Erstellt eine neue Sicherheitsgruppe, die dem Muster `DocumentDB-ec2-n` entspricht. Diese Sicherheitsgruppe hat eine eingehende Regel mit der VPC-Sicherheitsgruppe der EC2 Instance als Quelle. Diese Sicherheitsgruppe ist mit der Amazon DocumentDB DocumentDB-Datenbank verknüpft und ermöglicht der EC2 Instance den Zugriff auf die Amazon DocumentDB DocumentDB-Datenbank.
- Erstellt eine neue Sicherheitsgruppe, die dem Muster `ec2-DocumentDB-n` entspricht. Diese Sicherheitsgruppe hat eine ausgehende Regel mit der VPC-Sicherheitsgruppe der Amazon DocumentDB DocumentDB-Datenbank als Quelle. Diese Sicherheitsgruppe ist der Instance zugeordnet und ermöglicht der EC2 Instance, Traffic an die EC2 Amazon DocumentDB DocumentDB-Datenbank zu senden.

Amazon DocumentDB DocumentDB-Aktion: EC2 Sicherheitsgruppe zuordnen

Amazon DocumentDB ordnet die gültige, bestehende EC2 Sicherheitsgruppe der EC2 Instance zu. Diese Sicherheitsgruppe ermöglicht es der EC2 Instance, Traffic an die Amazon DocumentDB DocumentDB-Datenbank zu senden.

Anzeigen verbundener Rechenressourcen

Sie können den verwenden AWS Management Console , um die Rechenressourcen anzuzeigen, die mit einer Amazon DocumentDB DocumentDB-Datenbank verbunden sind. Zu den angezeigten Ressourcen gehören Rechenressourcenverbindungen, die automatisch eingerichtet wurden. Sie können die Konnektivität mit Rechenressourcen auf folgende Weise automatisch einrichten:

- Sie können die Rechenressource auswählen, wenn Sie die Datenbank erstellen. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster erstellen und Einen Multi-AZ-DB-Cluster erstellen.
- Sie können die Konnektivität zwischen einer vorhandenen Datenbank und einer Rechenressource einrichten. Weitere Informationen finden Sie unter Amazon EC2 automatisch Connect.

Die aufgelisteten Rechenressourcen enthalten keine Ressourcen, die manuell mit der Datenbank verbunden wurden. Sie können beispielsweise einer Rechenressource den manuellen Zugriff auf eine Datenbank erlauben, indem Sie der VPC-Sicherheitsgruppe, die der Datenbank zugeordnet ist, eine Regel hinzufügen.

Für die Auflistung einer Rechenressource müssen die folgenden Bedingungen erfüllt sei:

- Der Name der Sicherheitsgruppe, die der Rechenressource zugeordnet ist, entspricht dem Muster `ec2-DocumentDB-n` (wobei n eine Zahl ist).
- Die mit der Rechenressource verknüpfte Sicherheitsgruppe hat eine ausgehende Regel, bei der der Portbereich auf den Port festgelegt ist, den die Amazon DocumentDB DocumentDB-Datenbank verwendet.
- Die mit der Rechenressource verknüpfte Sicherheitsgruppe hat eine ausgehende Regel, bei der die Quelle auf eine Sicherheitsgruppe gesetzt ist, die der Amazon DocumentDB DocumentDB-Datenbank zugeordnet ist.
- Der Name der Sicherheitsgruppe, die mit der Amazon DocumentDB DocumentDB-Datenbank verknüpft ist, entspricht dem Muster `DocumentDB-ec2-n` (wobei n eine Zahl ist).
- Die mit der Amazon DocumentDB DocumentDB-Datenbank verknüpfte Sicherheitsgruppe hat eine Eingangsregel, bei der der Portbereich auf den Port festgelegt ist, den die Amazon DocumentDB DocumentDB-Datenbank verwendet.
- Die mit der Amazon DocumentDB DocumentDB-Datenbank verknüpfte Sicherheitsgruppe hat eine Eingangsregel, bei der die Quelle auf eine Sicherheitsgruppe gesetzt ist, die der Rechenressource zugeordnet ist.

Um Rechenressourcen anzuzeigen, die mit einer Amazon DocumentDB DocumentDB-Datenbank verbunden sind

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon DocumentDB DocumentDB-Konsole unter https://console.aws.amazon.com/docdb.
2. Wählen Sie im Navigationsbereich Datenbanken und dann den Namen der Amazon DocumentDB DocumentDB-Datenbank aus.
3. Sehen Sie sich auf der Registerkarte Konnektivität und Sicherheit die Rechenressourcen im Abschnitt Verbundene Rechenressourcen an.

Amazon EC2 manuell Connect

Themen

- Schritt 1: EC2 Amazon-Instance erstellen
- Schritt 2: Erstellen einer Sicherheitsgruppe
- Schritt 3: Erstellen Sie einen Amazon DocumentDB-Cluster

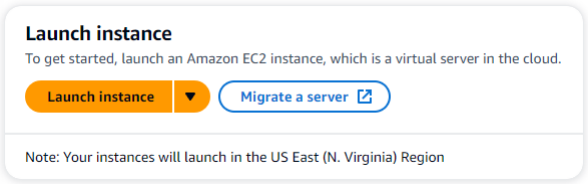
- Schritt 4: Connect zu Ihrer EC2 Amazon-Instance her
- Schritt 5: Installieren Sie die MongoDB-Shell
- Schritt 6: Amazon DocumentDB TLS verwalten
- Schritt 7: Connect zu Ihrem Amazon DocumentDB-Cluster her
- Schritt 8: Daten einfügen und abfragen
- Schritt 9: Erkunden

Bei den folgenden Schritten wird davon ausgegangen, dass Sie die Schritte im Voraussetzungen Thema abgeschlossen haben.

Schritt 1: EC2 Amazon-Instance erstellen

In diesem Schritt erstellen Sie eine EC2 Amazon-Instance in derselben Region und Amazon VPC, die Sie später für die Bereitstellung Ihres Amazon DocumentDB-Clusters verwenden werden.

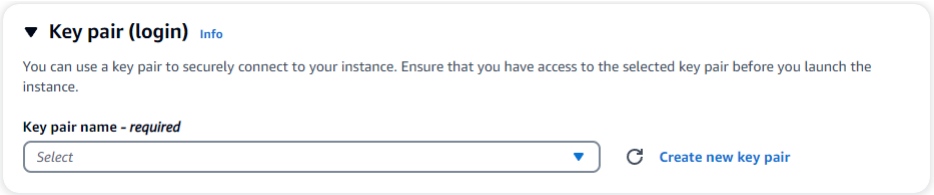
1. Wählen Sie auf der EC2 Amazon-Konsole Launch instance aus.



2. Geben Sie einen Namen oder eine Kennung in das Feld Name im Abschnitt Name und Tags ein.
3. Suchen Sie in der Dropdownliste Amazon Machine Image (AMI) nach Amazon Linux 2 AMI und wählen Sie es aus.



4. Suchen Sie in der Dropdownliste Instance-Typ nach t3.micro und wählen Sie es aus.
5. Geben Sie im Abschnitt key pair (Anmeldung) die Kennung eines vorhandenen Schlüsselpaars ein, oder wählen Sie Neues Schlüsselpaar erstellen.



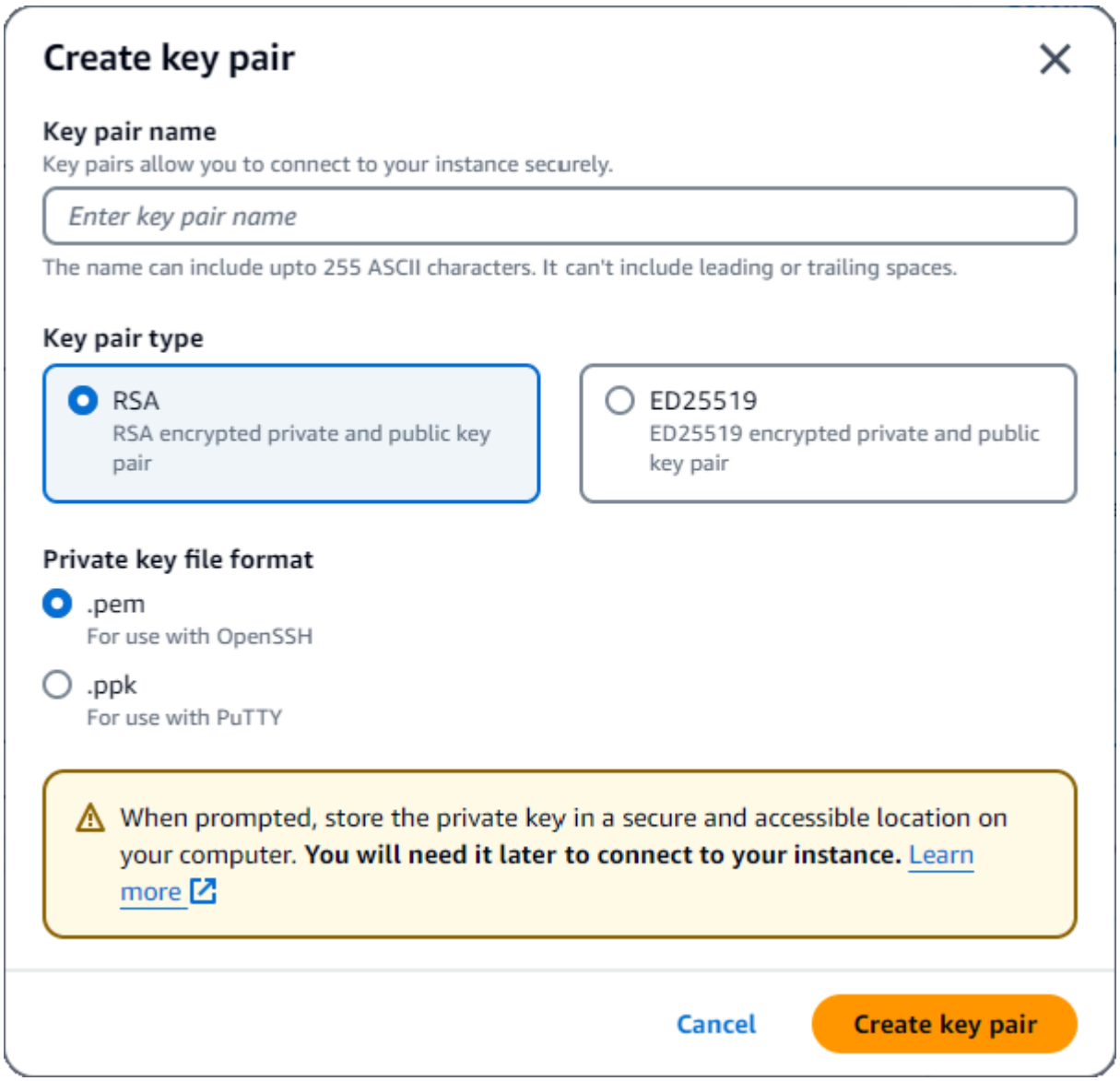
Sie müssen ein EC2 Amazon-Schlüsselpaar angeben.

Wenn Sie ein EC2 Amazon-Schlüsselpaar haben:

a. Wählen Sie ein key pair aus, wählen Sie Ihr key pair aus der Liste.
b. Sie müssen die private Schlüsseldatei (.pem- oder .ppk-Datei) bereits verfügbar haben, um sich bei Ihrer Amazon-Instance anmelden zu können. EC2

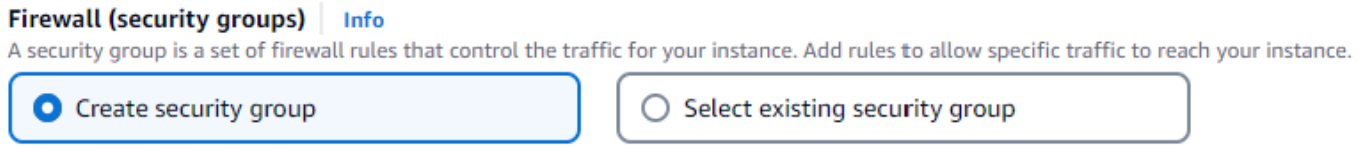
Wenn Sie kein EC2 Amazon-Schlüsselpaar haben:

a. Wählen Sie Neues key pair erstellen. Daraufhin wird das Dialogfeld key pair erstellen angezeigt.
b. Geben Sie einen Namen in das Feld Name des Schlüsselpaars ein.
c. Wählen Sie den Schlüsselpaartyp und das Dateiformat für den privaten Schlüssel.
d. Wählen Sie Create Key Pair (Schlüsselpaar erstellen) aus.



> **Note**
>
> Aus Sicherheitsgründen empfehlen wir dringend, ein Schlüsselpaar sowohl für die SSH- als auch für die Internetverbindung zu Ihrer Instance zu verwenden. EC2

6. Wählen Sie im Abschnitt Netzwerkeinstellungen unter Firewall (Sicherheitsgruppen) entweder Sicherheitsgruppe erstellen oder Bestehende Sicherheitsgruppe auswählen aus.



Wenn Sie eine bestehende Sicherheitsgruppe auswählen möchten, wählen Sie eine aus der Dropdownliste Allgemeine Sicherheitsgruppen aus.

Wenn Sie eine neue Sicherheitsgruppe erstellen möchten, gehen Sie wie folgt vor:

a. Überprüfen Sie alle Regeln für die Zulassung von Datenverkehr, die für Ihre EC2 Konnektivität gelten.

b. Wählen Sie im Feld IP die Option Meine IP oder Benutzerdefiniert aus, um aus einer Liste von CIDR-Blöcken, Präfixlisten oder Sicherheitsgruppen auszuwählen. Wir empfehlen Anywhere nicht als Wahl, es sei denn, Ihre EC2 Instance befindet sich in einem isolierten Netzwerk, da dort jede IP-Adresse Zugriff auf Ihre EC2 Instance hat.

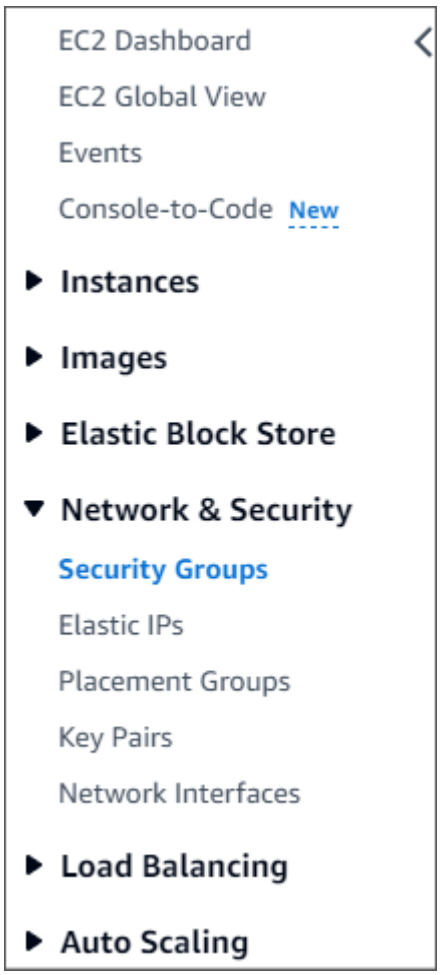


7. Überprüfen Sie im Abschnitt Zusammenfassung Ihre EC2 Konfiguration und wählen Sie, falls korrekt, die Option Launch instance aus.

Schritt 2: Erstellen einer Sicherheitsgruppe

Sie werden jetzt eine neue Sicherheitsgruppe in Ihrer Standard-Amazon-VPC erstellen. Die Sicherheitsgruppe `demoDocDB` ermöglicht es Ihnen, von Ihrer Amazon-Instance aus eine Verbindung zu Ihrem Amazon DocumentDB-Cluster auf Port 27017 (dem Standardport für Amazon DocumentDB) herzustellen. EC2

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



2. Wählen Sie Sicherheitsgruppe erstellen aus.



3. Gehen Sie im Abschnitt Grundlegende Details wie folgt vor:

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

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

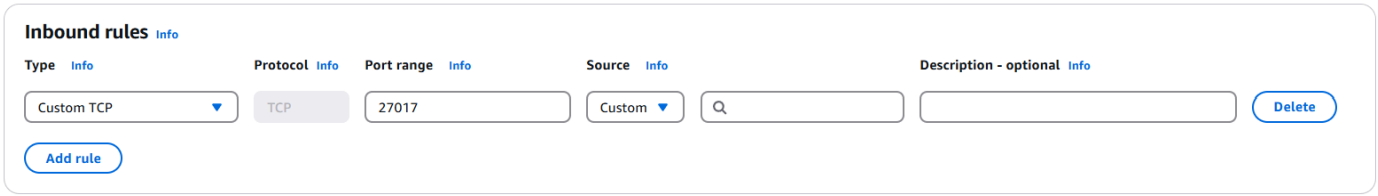
   c. Akzeptieren Sie für VPC die Verwendung Ihrer Standard-VPC.

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

   a. Wählen Sie für Typ die Option Benutzerdefinierte TCP-Regel (Standard) aus.

   b. Geben Sie für Portbereich den Wert ein`27017`.

   c. Wählen Sie für Source (Quelle) die Option Custom (Benutzerdefiniert) aus. Suchen Sie im Feld daneben nach der Sicherheitsgruppe, die Sie gerade in Schritt 1 erstellt haben. Möglicherweise müssen Sie Ihren Browser aktualisieren, damit die EC2 Amazon-Konsole den Quellnamen automatisch ausfüllt.



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



Schritt 3: Erstellen Sie einen Amazon DocumentDB-Cluster

Während die EC2 Amazon-Instance bereitgestellt wird, erstellen Sie Ihren Amazon DocumentDB-Cluster.

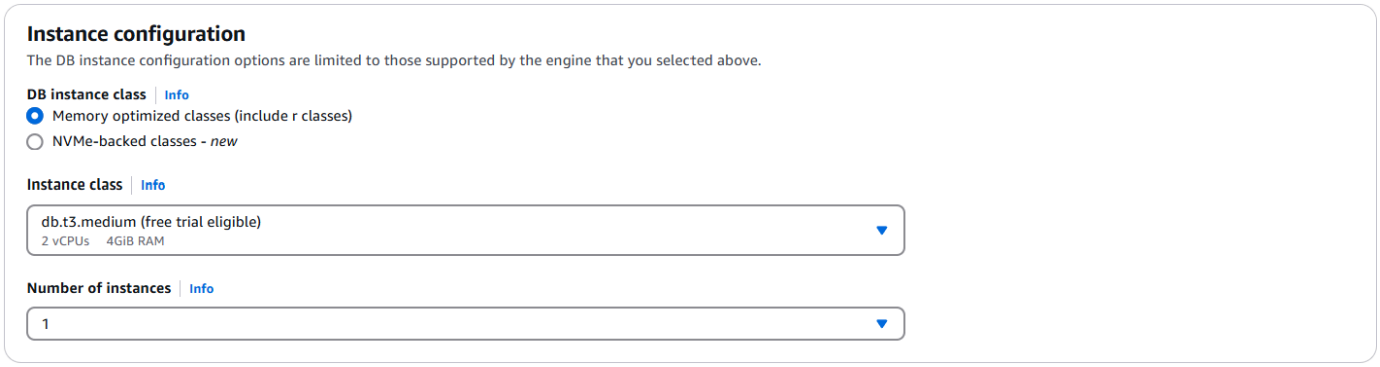
1. Navigieren Sie zur Amazon DocumentDB DocumentDB-Konsole und wählen Sie im Navigationsbereich Clusters aus.
2. Wählen Sie Erstellen aus.
3. Belassen Sie die Standardeinstellung für den Clustertyp auf Instance Based Cluster.
4. Geben Sie in Clusterkonfiguration für Cluster-ID einen eindeutigen Namen ein. Beachten Sie, dass die Konsole alle Clusternamen unabhängig davon, wie sie eingegeben wurden, in Kleinbuchstaben ändert.

   Behalten Sie für die Engine-Version den Standardwert 5.0.0 bei.
5. Behalten Sie für die Cluster-Speicherkonfiguration die Standardeinstellung von Amazon DocumentDB Standard bei.
6. In der Instanzkonfiguration:

   - Wählen Sie für DB-Instance-Klasse die Option Speicheroptimierte Klassen (einschließlich R-Klassen) aus (dies ist die Standardeinstellung).

     Bei der anderen Instance-Option handelt es sich um Klassen mit NVMe -backed. Weitere Informationen hierzu finden Sie unter NVMe-gestützte Instances.
   - Wählen Sie für Instance-Klasse den Instance-Typ aus, der Ihren Anforderungen entspricht. Eine detailliertere Erläuterung der Instanzklassen finden Sie unterSpezifikationen der Instanzklasse.

- Wählen Sie für die Anzahl der Instances eine Zahl aus, die Ihren Anforderungen am besten entspricht. Denken Sie daran, je niedriger die Anzahl, desto niedriger die Kosten und desto geringer das read/write Volumen, das vom Cluster verwaltet werden kann.



7. Behalten Sie für Konnektivität die Standardeinstellung „Keine Verbindung zu einer EC2 Rechenressource herstellen“ bei.

> **Note**
>
> Wenn Sie eine Verbindung zu einer EC2 Rechenressource herstellen, werden automatisch Sicherheitsgruppen für Ihre Verbindung zu Ihrem Cluster erstellt. Da Sie diese Sicherheitsgruppen im vorherigen Schritt manuell erstellt haben, sollten Sie „Keine Verbindung zu einer EC2 Rechenressource herstellen“ auswählen, um keinen zweiten Satz von Sicherheitsgruppen zu erstellen.

8. Geben Sie im Abschnitt Authentifizierung einen Benutzernamen für den Hauptbenutzer ein und wählen Sie dann Selbstverwaltet aus. Geben Sie ein Passwort ein und bestätigen Sie es.

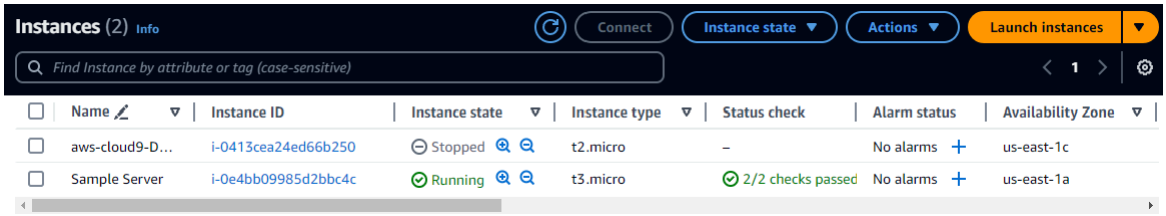
   Wenn Sie stattdessen Verwaltet in wählen AWS Secrets Manager, finden Sie Passwortverwaltung mit Amazon DocumentDB und AWS Secrets Manager weitere Informationen unter.

9. Wählen Sie Cluster erstellen.

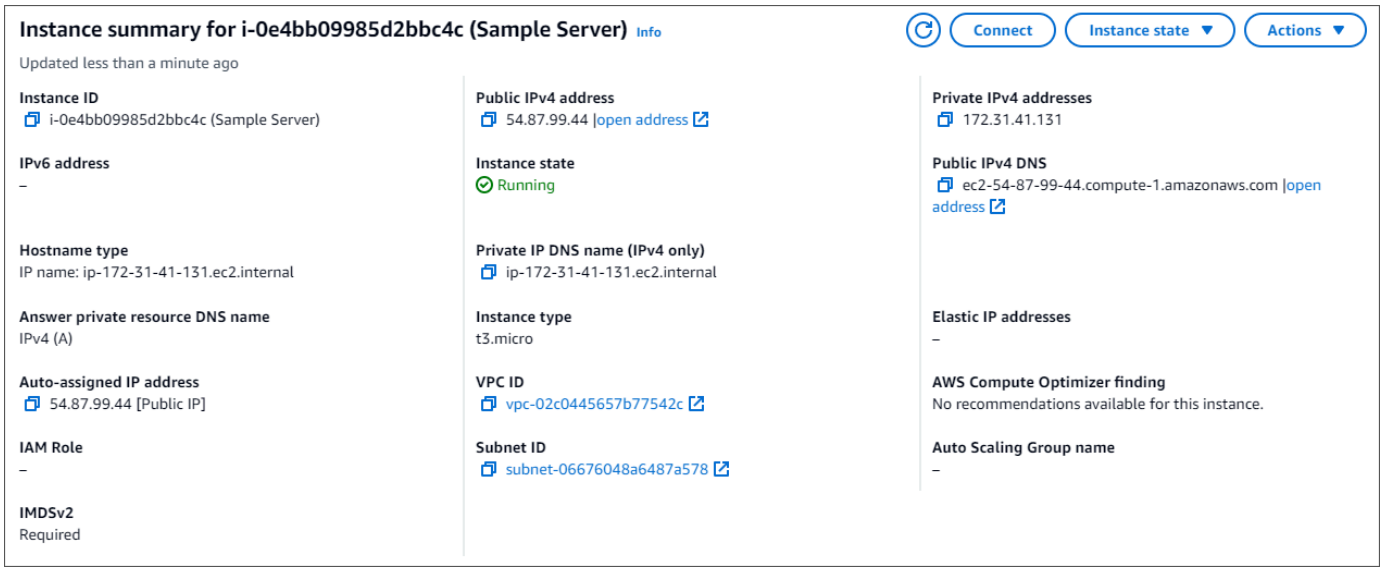
Schritt 4: Connect zu Ihrer EC2 Amazon-Instance her

Wenn Sie eine Verbindung zu Ihrer EC2 Amazon-Instance herstellen, können Sie die MongoDB-Shell installieren. Durch die Installation der Mongo-Shell können Sie eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen und ihn abfragen. Führen Sie folgende Schritte aus:

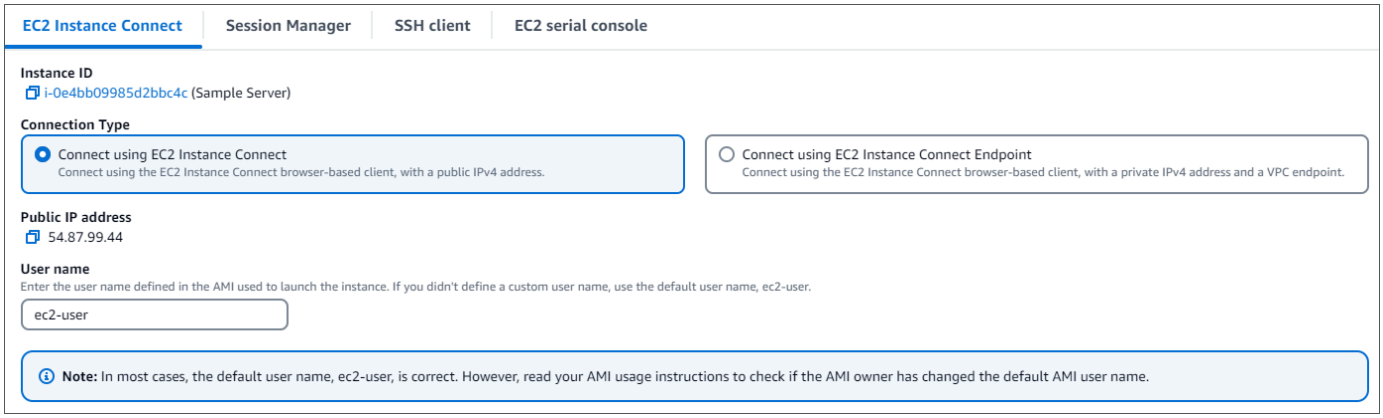
1. Navigieren Sie auf der EC2 Amazon-Konsole zu Ihren Instances und überprüfen Sie, ob die gerade erstellte Instance läuft. Wenn ja, wählen Sie die Instance aus, indem Sie auf die Instance-ID klicken.



2. Wählen Sie Connect aus.



3. Es gibt vier Optionen mit Registerkarten für Ihre Verbindungsmethode: Amazon EC2 Instance Connect, Session Manager, SSH-Client oder EC2 serielle Konsole. Sie müssen eine auswählen und deren Anweisungen befolgen. Wenn Sie fertig sind, wählen Sie Connect.



> **Note**
>
> Wenn sich Ihre IP-Adresse geändert hat, nachdem Sie diesen Walkthrough gestartet haben, oder wenn Sie zu einem späteren Zeitpunkt zu Ihrer Umgebung zurückkehren, müssen Sie Ihre demoEC2 Sicherheitsgruppenregel für eingehenden Datenverkehr aktualisieren, um eingehenden Datenverkehr von Ihrer neuen API-Adresse aus zu ermöglichen.

Schritt 5: Installieren Sie die MongoDB-Shell

Sie können jetzt die MongoDB-Shell installieren. Dabei handelt es sich um ein Befehlszeilenprogramm, mit dem Sie eine Verbindung herstellen und Ihren Amazon DocumentDB-Cluster abfragen. Derzeit gibt es zwei Versionen der MongoDB-Shell: die neueste Version, Mongosh, und die vorherige Version, Mongo-Shell.

> **Important**
>
> Es gibt eine bekannte Einschränkung bei Treibern von Node.js, die älter als Version 6.13.1 sind und die derzeit nicht von der IAM-Identitätsauthentifizierung für Amazon DocumentDB unterstützt werden. Die Treiber und Tools der Datei Node.js, die den Treiber Node.js verwenden (z. B. mongosh), müssen aktualisiert werden, sodass sie den Treiber Node.js der Version 6.13.1 oder höher verwenden.

Folgen Sie den nachstehenden Anweisungen, um die MongoDB-Shell für Ihr Betriebssystem zu installieren.

On Amazon Linux

So installieren Sie die MongoDB-Shell auf Amazon Linux

Wenn Sie IAM nicht verwenden und die neueste MongoDB-Shell (mongosh) verwenden möchten, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, gehen Sie wie folgt vor:

1. Erstellen Sie die Repository-Datei. Führen Sie in der Befehlszeile Ihrer EC2 Instanz, die Sie erstellt haben, den folgenden Befehl aus:

```
echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
repo.mongodb.org/yum/amazon/2023/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
```

```
 \ngpgkey=https://pgp.mongodb.com/server-5.0.asc" | sudo tee /etc/yum.repos.d/
mongodb-org-5.0.repo
```

2. Wenn der Vorgang abgeschlossen ist, installieren Sie mongosh mit einer der beiden folgenden Befehlsoptionen an der Befehlszeile:

   Option 1 — Wenn Sie während der EC2 Amazon-Konfiguration das Standard-Amazon Linux 2023 ausgewählt haben, geben Sie diesen Befehl ein:

   ```
   sudo yum install -y mongodb-mongosh-shared-openssl3
   ```

   Option 2 — Wenn Sie bei der EC2 Amazon-Konfiguration Amazon Linux 2 ausgewählt haben, geben Sie diesen Befehl ein:

   ```
   sudo yum install -y mongodb-mongosh
   ```

Wenn Sie IAM verwenden, müssen Sie die vorherige Version der MongoDB-Shell (5.0) verwenden, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen. Gehen Sie wie folgt vor:

1. Erstellen Sie die Repository-Datei. Führen Sie in der Befehlszeile Ihrer EC2 Instanz, die Sie erstellt haben, den folgenden Befehl aus:

   ```
   echo -e "[mongodb-org-5.0] \nname=MongoDB Repository\nbaseurl=https://
   repo.mongodb.org/yum/amazon/2023/mongodb-org/5.0/x86_64/\ngpgcheck=1 \nenabled=1
    \ngpgkey=https://pgp.mongodb.com/server-5.0.asc" | sudo tee /etc/yum.repos.d/
   mongodb-org-5.0.repo
   ```

2. Wenn der Vorgang abgeschlossen ist, installieren Sie die Mongodb 5.0-Shell mit der folgenden Befehlsoption an der Befehlszeile:

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

On Ubuntu

Um mongosh auf Ubuntu zu installieren

1. Importieren Sie den öffentlichen Schlüssel, der von dem Paketverwaltungssystem verwendet wird.

```
curl -fsSL https://pgp.mongodb.com/server-5.0.asc | sudo gpg --dearmor -o /usr/
share/keyrings/mongodb-server-5.0.gpg
```

2. Erstellen Sie die Listendatei `mongodb-org-5.0.list` für MongoDB mit dem korrekten Befehl für Ihre Ubuntu-Version.

```
echo "deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-
server-5.0.gpg ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0
 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list
```

3. Importieren und aktualisieren Sie die lokale Paketdatenbank mit dem folgenden Befehl:

```
sudo apt-get update
```

4. Installieren Sie Mongosh.

```
sudo apt-get install -y mongodb-mongosh
```

Weitere Informationen zum Installieren von früheren Versionen von MongoDB auf Ihrem Ubuntu-System finden Sie unter Installieren von MongoDB Community Edition auf Ubuntu.

On other operating systems

Informationen zum Installieren der mongo-Shell auf anderen Betriebssystemen finden Sie unter Installieren von MongoDB Community Edition in der MongoDB-Dokumentation.

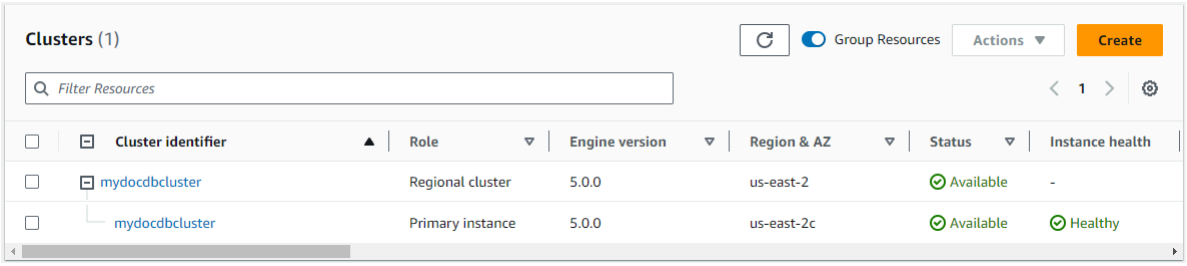
Schritt 6: Amazon DocumentDB TLS verwalten

Laden Sie das CA-Zertifikat für Amazon DocumentDB mit dem folgenden Code herunter: `wget https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem`

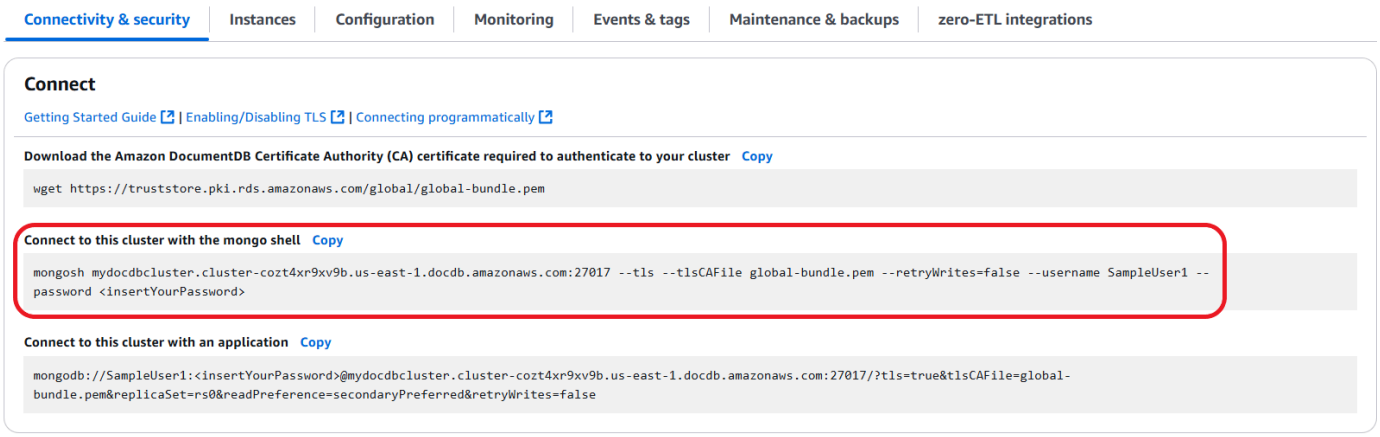
> **Note**
>
> Transport Layer Security (TLS) ist standardmäßig für alle neuen Amazon DocumentDB-Cluster aktiviert. Weitere Informationen finden Sie unter TLS-Einstellungen für Amazon DocumentDB-Cluster verwalten.

Schritt 7: Connect zu Ihrem Amazon DocumentDB-Cluster her

1. Suchen Sie auf der Amazon DocumentDB DocumentDB-Konsole unter Clusters Ihren Cluster. Wählen Sie den Cluster aus, den Sie erstellt haben, indem Sie auf die Cluster-ID für diesen Cluster klicken.



2. Suchen Sie auf der Registerkarte Konnektivität und Sicherheit im Feld Connect nach Connect to this cluster with the mongo shell:



   Kopieren Sie die angegebene Verbindungszeichenfolge und fügen Sie sie in Ihr Terminal ein.

   Nehmen Sie die folgenden Änderungen daran vor:

   a. Stellen Sie sicher, dass Sie den richtigen Benutzernamen in der Zeichenfolge haben.

   b. Auslassen, `<insertYourPassword>` sodass Sie beim Herstellen der Verbindung von der Mongo-Shell zur Eingabe des Passworts aufgefordert werden.

c. Optional: Wenn Sie die IAM-Authentifizierung oder die vorherige Version der MongoDB-Shell verwenden, ändern Sie Ihre Verbindungszeichenfolge wie folgt:

```
mongo --ssl --host mydocdbcluster.cluster-cozt4xr9xv9b.us-
east-1.docdb.amazonaws.com:27017 --sslCAFile global-bundle.pem --
username SampleUser1 --password
```

`mydocdbcluster.cluster-cozt4xr9xv9b.us-east-1`Ersetzen Sie durch dieselben Informationen aus Ihrem Cluster.

3. Drücken Sie die Eingabetaste in Ihrem Terminal. Sie werden nun zur Eingabe Ihres Passworts aufgefordert. Geben Sie Ihr Passwort ein.
4. Wenn Sie Ihr Passwort eingeben und die `rs0 [direct: primary] <env-name>>` Aufforderung sehen, sind Sie erfolgreich mit Ihrem Amazon DocumentDB-Cluster verbunden.

Haben Sie Probleme beim Herstellen einer Verbindung? Weitere Informationen finden Sie unter Fehlerbehebung bei Amazon DocumentDB.

Schritt 8: Daten einfügen und abfragen

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

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

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

Sie erhalten die folgende Ausgabe:

```
{
  acknowledged: true,
  insertedId: ObjectId('673657216bdf6258466b128c')
}
```

2. Sie können das Dokument lesen, das Sie mit dem `findOne()` Befehl geschrieben haben (weil er nur ein einziges Dokument zurückgibt). Geben Sie Folgendes ein:

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

Sie erhalten die folgende Ausgabe:

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

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

```
db.profiles.insertMany([{ _id: 1, name: 'Matt', status: 'active', level: 12, score:
 202 },
      { _id: 2, name: 'Frank', status: 'inactive', level: 2, score: 9 },
      { _id: 3, name: 'Karen', status: 'active', level: 7, score: 87 },
      { _id: 4, name: 'Katie', status: 'active', level: 3, score: 27 }
])
```

Sie erhalten die folgende Ausgabe:

```
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3, '3': 4 } }
```

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

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

Sie erhalten eine Ausgabe, die den in Schritt 3 eingegebenen Daten entspricht.

5. Verwenden Sie mithilfe eines Filters eine Abfrage für ein einzelnes Dokument. Geben Sie Folgendes ein:

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

Sie erhalten die folgende Ausgabe:

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

6. Versuchen wir nun, ein Profil zu finden und es mit dem `findAndModify` Befehl zu ändern. Wir geben dem Benutzer Matt zusätzliche 10 Punkte mit dem folgenden Code:

```
db.profiles.findAndModify({
```

```
        query: { name: "Matt", status: "active"},
        update: { $inc: { score: 10 } }
    })
```

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

```
{
    [{_id : 1, name : 'Matt', status: 'active', level: 12, score: 202}]
```

7. Sie können mit der folgenden Abfrage überprüfen, ob sich sein Punktestand geändert hat:

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

   Sie erhalten die folgende Ausgabe:

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

Schritt 9: Erkunden

Herzlichen Glückwunsch! Sie haben die Schnellstartanleitung für Amazon DocumentDB erfolgreich abgeschlossen.

Was kommt als Nächstes? Erfahren Sie, wie Sie diese leistungsstarke Datenbank mit einigen ihrer beliebten Funktionen voll ausschöpfen können:

- Amazon DocumentDB verwalten
- Skalierung
- Sichern und Wiederherstellen

> **Note**
>
> Um Kosten zu sparen, können Sie entweder Ihren Amazon DocumentDB-Cluster stoppen, um die Kosten zu senken, oder den Cluster löschen. Standardmäßig stoppt Ihre AWS Cloud9 Umgebung nach 30 Minuten Inaktivität die zugrunde liegende EC2 Amazon-Instance.

Stellen Sie mithilfe des Amazon DocumentDB DocumentDB-JDBC-Treibers eine Verbindung her

Der JDBC-Treiber für Amazon DocumentDB bietet Entwicklern eine relationale SQL-Schnittstelle und ermöglicht die Konnektivität von BI-Tools wie Tableau und. DbVisualizer

Weitere Informationen finden Sie in der Amazon DocumentDB DocumentDB-JDBC-Treiber-Dokumentation unter. GitHub

Themen

- Erste Schritte
- Stellen Sie von Tableau Desktop aus eine Connect zu Amazon DocumentDB her
- Connect zu Amazon DocumentDB her von DbVisualizer
- Automatische JDBC-Schemagenerierung
- SQL-Unterstützung und Einschränkungen
- Fehlerbehebung

Erste Schritte

Schritt 1. Amazon DocumentDB-Cluster erstellen

Wenn Sie keinen Amazon DocumentDB-Cluster erstellt haben, erstellen Sie einen anhand der Anweisungen im Abschnitt Erste Schritte im Amazon DocumentDB DocumentDB-Entwicklerhandbuch.

> **Note**
>
> Amazon DocumentDB ist ein reiner Virtual Private Cloud (VPC) -Service. Wenn Sie eine Verbindung von einem lokalen Computer außerhalb der VPC des Clusters herstellen, müssen Sie eine SSH-Verbindung zu einer EC2 Amazon-Instance herstellen. Starten Sie in diesem Fall Ihren Cluster mithilfe der Anweisungen unter Connect with EC2. Weitere Informationen zum SSH-Tunneling und wann Sie es möglicherweise benötigen, finden Sie unter Verwenden eines SSH-Tunnels zum Herstellen einer Connect zu Amazon DocumentDB.

Schritt 2. JRE- oder JDK-Installation

Abhängig von Ihrer BI-Anwendung müssen Sie möglicherweise sicherstellen, dass eine 64-Bit-JRE- oder JDK-Installationsversion 8 oder höher auf Ihrem Computer installiert ist. Sie können das Java SE Runtime Environment 8 hier herunterladen.

Schritt 3. Laden Sie den DocumentDB-JDBC-Treiber herunter

Laden Sie den DocumentDB-JDBC-Treiber von hier herunter. Der Treiber ist als einzelne JAR-Datei verpackt (z. B. documentdb-jdbc-1.0.0-all.jar).

Schritt 4. Verwenden eines SSH-Tunnels zum Connect zu Amazon DocumentDB

Amazon DocumentDB-Cluster (mit MongoDB-Kompatibilität) werden in einer Amazon Virtual Private Cloud (Amazon VPC) bereitgestellt. Sie können direkt von EC2 Amazon-Instances oder anderen AWS Services abgerufen werden, die in derselben Amazon VPC bereitgestellt werden. Darüber hinaus können EC2a Instances oder andere AWS Services VPCs in verschiedenen Regionen derselben AWS Region oder anderen Regionen über VPC-Peering auf Amazon DocumentDB zugreifen.

Sie können SSH-Tunneling (auch bekannt als Portweiterleitung) verwenden, um von außerhalb der VPC des Clusters auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen. Dies wird bei den meisten Benutzern der Fall sein, die ihre Anwendung nicht auf einer VM in derselben VPC wie der DocumentDB-Cluster ausführen.

Um einen SSH-Tunnel zu erstellen, benötigen Sie eine EC2 Amazon-Instance, die in derselben Amazon VPC wie Ihr Amazon DocumentDB-Cluster läuft. Sie können entweder eine bestehende EC2 Instanz in derselben VPC wie Ihr Cluster verwenden oder eine erstellen. Sie können einen SSH-Tunnel zum Amazon DocumentDB-Cluster einrichten, `sample-cluster.node.us-east-1.docdb.amazonaws.com` indem Sie den folgenden Befehl auf Ihrem lokalen Computer ausführen.

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -
N
```

Das Flag -L wird für die Weiterleitung eines lokalen Ports verwendet. Dies ist eine Voraussetzung für die Verbindung zu einem BI-Tool, das auf einem Client außerhalb Ihrer VPC ausgeführt wird. Sobald Sie den obigen Schritt ausgeführt haben, können Sie mit den nächsten Schritten für das BI-Tool Ihrer Wahl fortfahren.

Weitere Informationen zum SSH-Tunneling finden Sie in der Dokumentation zum Herstellen einer Verbindung mit Amazon DocumentDB über einen SSH-Tunnel.

Stellen Sie von Tableau Desktop aus eine Connect zu Amazon DocumentDB her

Themen

- Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers
- Mit Tableau eine Verbindung zu Amazon DocumentDB herstellen — SSH-Tunnel

Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers

Um von Tableau Desktop aus eine Verbindung zu Amazon DocumentDB herzustellen, müssen Sie den Amazon DocumentDB DocumentDB-JDBC-Treiber und den DocumentDB-Tableau-Connector herunterladen und installieren.

1. Laden Sie die JAR-Datei des Amazon DocumentDB DocumentDB-JDBC-Treibers aus dem Amazon DocumentDB DocumentDB-JDBC-Treiber-Repository herunter und kopieren Sie sie je nach Betriebssystem in eines der folgenden Verzeichnisse:
   - Windows - `C:\Program Files\Tableau\Drivers`
   - macOS - `~/Library/Tableau/Drivers`
2. Laden Sie den DocumentDB Tableau-Connector (eine TACO-Datei) von der Tableau Exchange-Website herunter und kopieren Sie ihn in Ihr My Tableau Repository/Connectors-Verzeichnis.
   - Windows - `C:\Users\[user]\Documents\My Tableau Repository\Connectors`
   - macOS - `/Users/[user]/Documents/My Tableau Repository/Connectors`

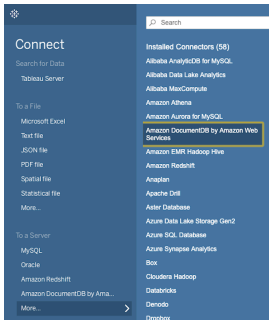
Weitere Informationen finden Sie in der Tableau-Dokumentation.

> **Note**
>
> Wenn Sie neuere CA-Zertifikate verwenden, stellen Sie sicher, dass Sie Ihren JDBC-Treiber auf Version 1.4.5 aktualisieren (in diesem AWS GitHub Repository verfügbar).

Mit Tableau eine Verbindung zu Amazon DocumentDB herstellen — SSH-Tunnel

Um von einem Client-Computer außerhalb der VPC Ihres DocumentDB-Clusters aus eine Verbindung zu Tableau herzustellen, müssen Sie einen SSH-Tunnel einrichten, bevor Sie die folgenden Schritte ausführen:

1. Starten Sie die Tableau Desktop-Anwendung.
2. Navigieren Sie zu Connect > Zu einem Server > Mehr.
3. Wählen Sie unter Installierte Konnektoren die Option Amazon DocumentDB by Amazon Web Services aus.

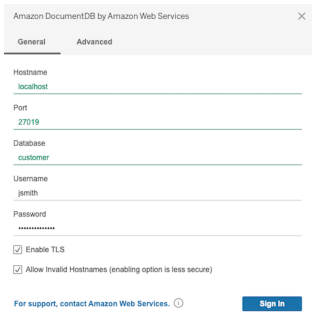


Herstellen einer Verbindung zu Amazon DocumentDB mithilfe von Tableau — Externer SSH-Tunnel

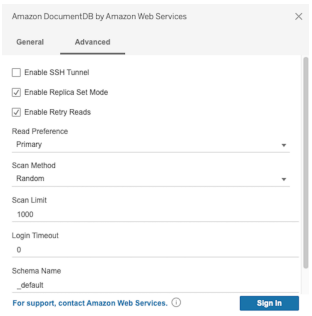
1. Geben Sie die erforderlichen Verbindungsparameter Hostname, Port, Datenbank, Benutzername und Passwort ein. Die Verbindungsparameter im folgenden Beispiel entsprechen der JDBC-Verbindungszeichenfolge:

   ```
   jdbc:documentdb://localhost:27019/test?
   tls=true&tlsAllowInvalidHostnames=true&scanMethod=random&scanLimit=1000&logir
   ```

   die Parameter Benutzername und Passwort separat in einer Eigenschaftensammlung übergeben werden. Weitere Informationen zu Verbindungszeichenfolgenparametern finden Sie in der Github-Dokumentation des Amazon DocumentDB DocumentDB-JDBC-Treibers.



2. (Optional) Weitere erweiterte Optionen finden Sie auf der Registerkarte Erweitert.

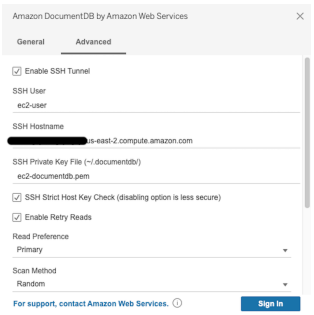


3. Klicken Sie auf Sign in.

Mit Tableau eine Verbindung zu Amazon DocumentDB herstellen — Interner SSH-Tunnel

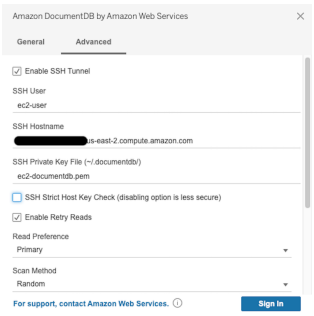
> **Note**
>
> Wenn Sie es vorziehen, den SSH-Tunnel nicht über ein Terminal einzurichten, können Sie die Tableau-GUI verwenden, um Ihre EC2 Instanzdetails anzugeben, die der JDBC-Treiber standardmäßig zur Erstellung eines SSH-Tunnels verwenden wird.

1. Wählen Sie auf der Registerkarte Erweitert die Option SSH-Tunnel aktivieren aus, um weitere Eigenschaften zu überprüfen.



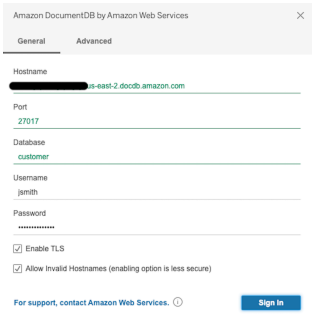
2. Geben Sie den SSH-Benutzer, den SSH-Hostnamen und die private SSH-Schlüsseldatei ein.
3. (Optional) Sie können die Option SSH Strict Host Key Check deaktivieren, wodurch die Überprüfung des Hostschlüssels anhand einer Datei mit bekannten Hosts umgangen wird.

> **Note**
>
> Die Deaktivierung dieser Option ist weniger sicher, da sie zu einem Angriff führen kann. man-in-the-middle



4. Geben Sie die erforderlichen Parameter ein: Hostname, Port, Datenbank, Benutzername und Passwort.

> **Note**
>
> Stellen Sie sicher, dass Sie den DocumentDB-Clusterendpunkt und nicht localhost verwenden, wenn Sie die interne SSH-Tunneloption verwenden.



5. Klicken Sie auf Sign In.

Connect zu Amazon DocumentDB her von DbVisualizer

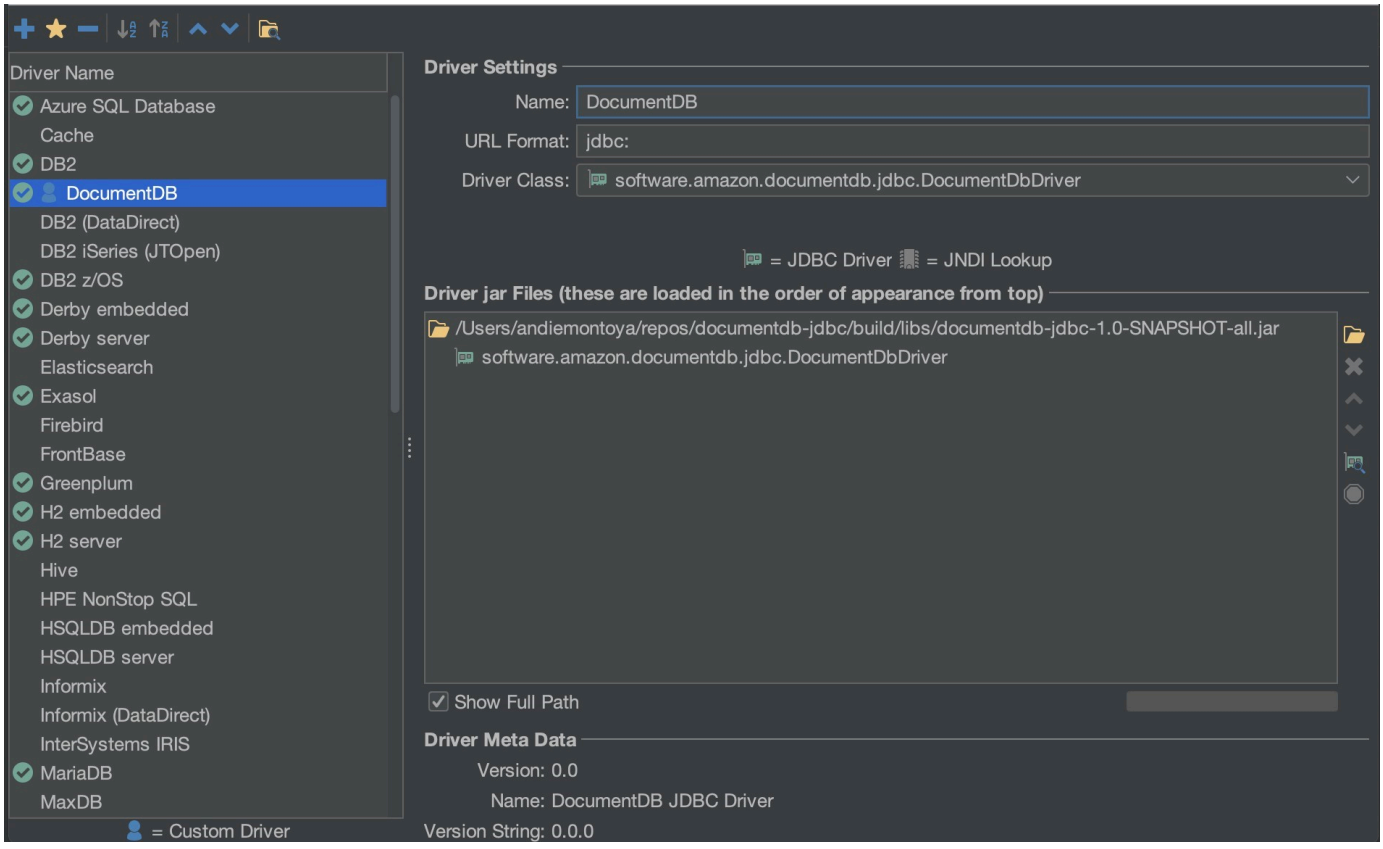
Themen

- Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers
- Verbindung zu Amazon DocumentDB herstellen mit DbVisualizer

Hinzufügen des Amazon DocumentDB DocumentDB-JDBC-Treibers

Um eine Verbindung zu Amazon DocumentDB herzustellen, müssen DbVisualizer Sie zuerst den Amazon DocumentDB DocumentDB-JDBC-Treiber importieren.

1. Starten Sie die DbVisualizer Anwendung und navigieren Sie zum Menüpfad: Tools > Driver Manager...
2. Wählen Sie + (oder wählen Sie im Menü Treiber > Treiber erstellen).
3. Legen Sie Name auf `DocumentDB` fest.
4. Stellen Sie das URL-Format ein auf `jdbc:documentdb://<host>[:port]/<database>[?option=value[&option=value[...]]]`
5. Wählen Sie die Ordnerschaltfläche und dann die JAR-Datei des Amazon DocumentDB DocumentDB-JDBC-Treibers aus und klicken Sie auf die Schaltfläche Öffnen.
6. Stellen Sie sicher, dass das Feld Treiberklasse auf eingestellt ist. `software.amazon.documentdb.jdbc.DocumentDbDriver` Ihre Driver Manager-Einstellungen für DocumentDB sollten wie im folgenden Beispiel aussehen.



7. Schließen Sie das Dialogfeld. Der Amazon DocumentDB DocumentDB-JDBC-Treiber wird eingerichtet und ist einsatzbereit.

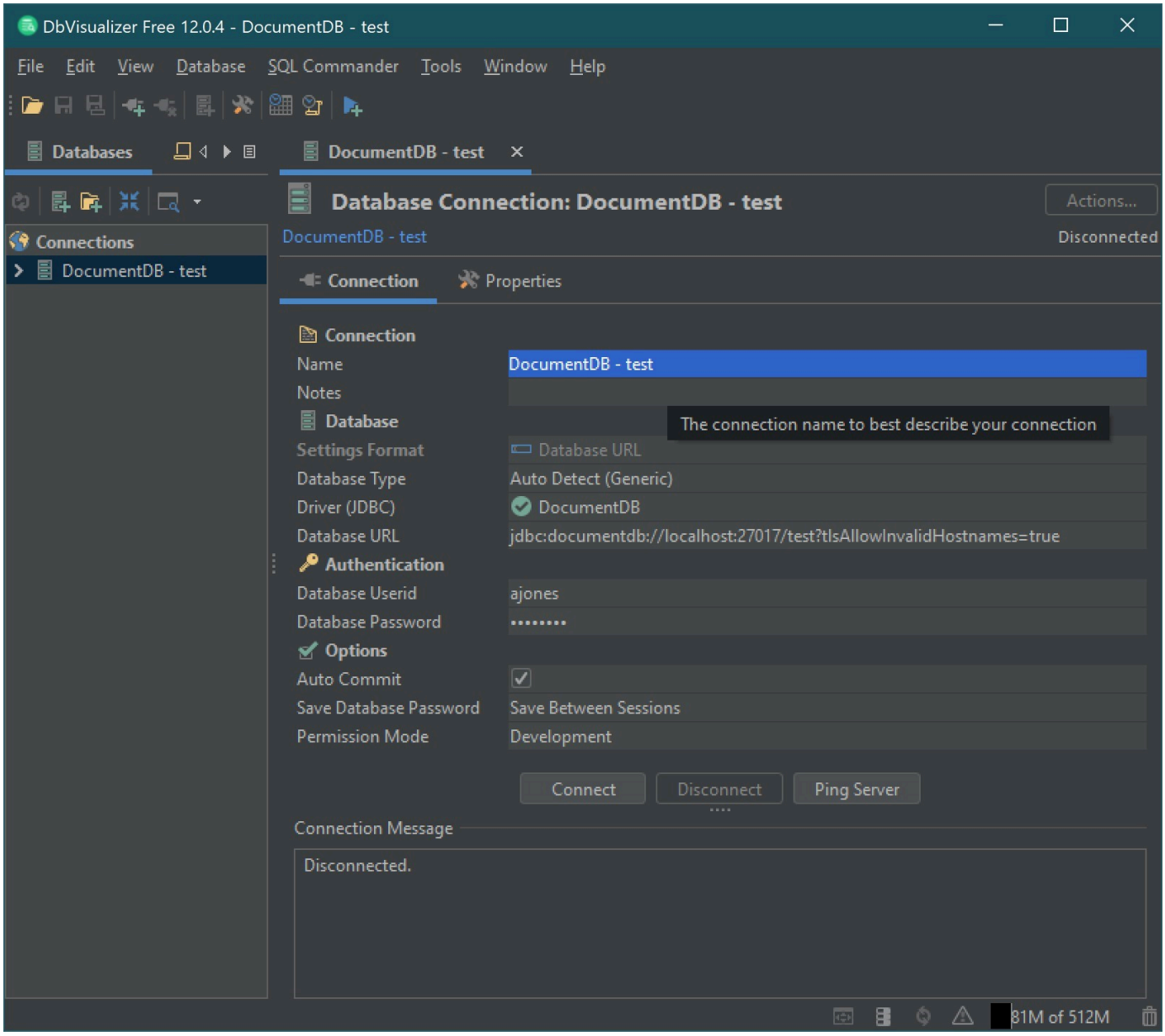
Verbindung zu Amazon DocumentDB herstellen mit DbVisualizer

Stellen Sie mithilfe von Amazon DocumentDB eine Connect DbVisualizer

1. Wenn Sie von außerhalb der VPC des Amazon DocumentDB-Clusters eine Verbindung herstellen, stellen Sie sicher, dass Sie einen SSH-Tunnel eingerichtet haben.
2. Wählen Sie im Menü der obersten Ebene Datenbank > Datenbankverbindung erstellen.
3. Geben Sie einen aussagekräftigen Namen für das Feld „Name“ ein.
4. Stellen Sie Driver (JDBC) auf den DocumentDB-Treiber ein, den Sie im vorherigen Abschnitt erstellt haben.
5. Stellen Sie die Datenbank-URL auf Ihre JDBC-Verbindungszeichenfolge ein.

   Zum Beispiel: `jdbc:documentdb://localhost:27017/database?tlsAllowInvalidHostnames=true`
6. Setzen Sie Database Userid auf Ihre Amazon DocumentDB DocumentDB-Benutzer-ID.
7. Stellen Sie das Datenbankpasswort auf das entsprechende Passwort für die Benutzer-ID ein.

   Ihr Datenbankverbindungsdialog sollte wie der folgende Dialog aussehen:



8. Wählen Sie Connect aus.

Automatische JDBC-Schemagenerierung

Amazon DocumentDB ist eine Dokumentendatenbank und hat daher nicht das Konzept von Tabellen und Schemas. BI-Tools wie Tableau erwarten jedoch, dass die Datenbank, mit der sie eine Verbindung herstellen, ein Schema darstellt. Insbesondere wenn die JDBC-Treiberverbindung das Schema für die Sammlung in der Datenbank abrufen muss, fragt sie nach allen Sammlungen in der Datenbank ab. Der Treiber ermittelt, ob bereits eine zwischengespeicherte Version des Schemas für diese Sammlung vorhanden ist. Wenn keine zwischengespeicherte Version vorhanden ist, wird

anhand der Sammlung nach Dokumenten gesucht und ein Schema erstellt, das auf dem folgenden Verhalten basiert.

Themen

- Einschränkungen bei der Schemagenerierung
- Optionen für die Scanmethode
- Amazon-DocumentDB-Datentypen
- Zuordnung skalarer Dokumentfelder
- Behandlung von Objekt- und Array-Datentypen

Einschränkungen bei der Schemagenerierung

Der DocumentDB-JDBC-Treiber begrenzt die Länge von Bezeichnern auf 128 Zeichen. Der Schema-Generator kann die Länge der generierten Bezeichner (Tabellennamen und Spaltennamen) kürzen, um sicherzustellen, dass sie dieser Grenze entsprechen.

Optionen für die Scanmethode

Das Sampling-Verhalten kann mithilfe von Verbindungszeichenfolgen- oder Datenquellenoptionen geändert werden.

- ScanMethod= <option>
  - random — (Standard) — Die Beispieldokumente werden in zufälliger Reihenfolge zurückgegeben.
  - idForward — Die Beispieldokumente werden in der Reihenfolge ihrer ID zurückgegeben.
  - idReverse — Die Beispieldokumente werden in umgekehrter Reihenfolge der ID zurückgegeben.
  - all — Probieren Sie alle Dokumente in der Sammlung aus.
- scanLimit= <n>— Die Anzahl der Dokumente, die gesampelt werden sollen. Der Wert muss eine positive ganze Zahl sein. Der Standardwert ist 1000. Wenn ScanMethod auf all gesetzt ist, wird diese Option ignoriert.

Amazon-DocumentDB-Datentypen

Der Amazon DocumentDB-Server unterstützt eine Reihe von MongoDB-Datentypen. Im Folgenden sind die unterstützten Datentypen und die zugehörigen JDBC-Datentypen aufgeführt.

MongoDB-Datentyp	Wird in DocumentDB unterstützt	JDBC-Datentyp
Binäre Daten	Ja	VARBINARY
Boolesch	Ja	BOOLEAN
Double	Ja	DOUBLE
32-Bit-Ganzzahl	Ja	INTEGER
64-Bit-Ganzzahl	Ja	BIGINT
String	Ja	VARCHAR
ObjectId	Ja	VARCHAR
Datum	Ja	TIMESTAMP (ZEITSTEMPEL)
Null	Ja	VARCHAR
Regulärer Ausdruck	Ja	VARCHAR
Zeitstempel	Ja	VARCHAR
MinKey	Ja	VARCHAR
MaxKey	Ja	VARCHAR
Object	Ja	virtuelle Tabelle
Array	Ja	virtueller Tisch
Decimal128	Nein	DECIMAL
JavaScript	Nein	VARCHAR
JavaScript (mit Umfang)	Nein	VARCHAR
Undefined	Nein	VARCHAR
Symbol	Nein	VARCHAR

MongoDB-Datentyp	Wird in DocumentDB unterstützt	JDBC-Datentyp
DBPointer (4.0+)	Nein	VARCHAR

Zuordnung skalarer Dokumentfelder

Beim Scannen einer Stichprobe von Dokumenten aus einer Sammlung erstellt der JDBC-Treiber ein oder mehrere Schemas, um die Beispiele in der Sammlung darzustellen. Im Allgemeinen wird ein Skalarfeld im Dokument einer Spalte im Tabellenschema zugeordnet. In einer Sammlung mit dem Namen Team und einem einzelnen Dokument würde dies `{ "_id" : "112233", "name" : "Alastair", "age": 25 }` beispielsweise dem Schema entsprechen:

Tabellenname	Spaltenname	Datentyp	Schlüssel
Mannschaft	Team-ID	VARCHAR	PK
Mannschaft	Name	VARCHAR	
Mannschaft	age	INTEGER	

Förderung von Konflikten beim Datentyp

Beim Scannen der Musterdokumente ist es möglich, dass die Datentypen für ein Feld von Dokument zu Dokument nicht einheitlich sind. In diesem Fall stuft der JDBC-Treiber den JDBC-Datentyp auf einen gemeinsamen Datentyp herauf, der für alle Datentypen aus den Stichprobendokumenten geeignet ist.

Beispiel:

```
{
"_id" : "112233",
"name" : "Alastair", "age" : 25
}

{
"_id" : "112244",
"name" : "Benjamin",
```

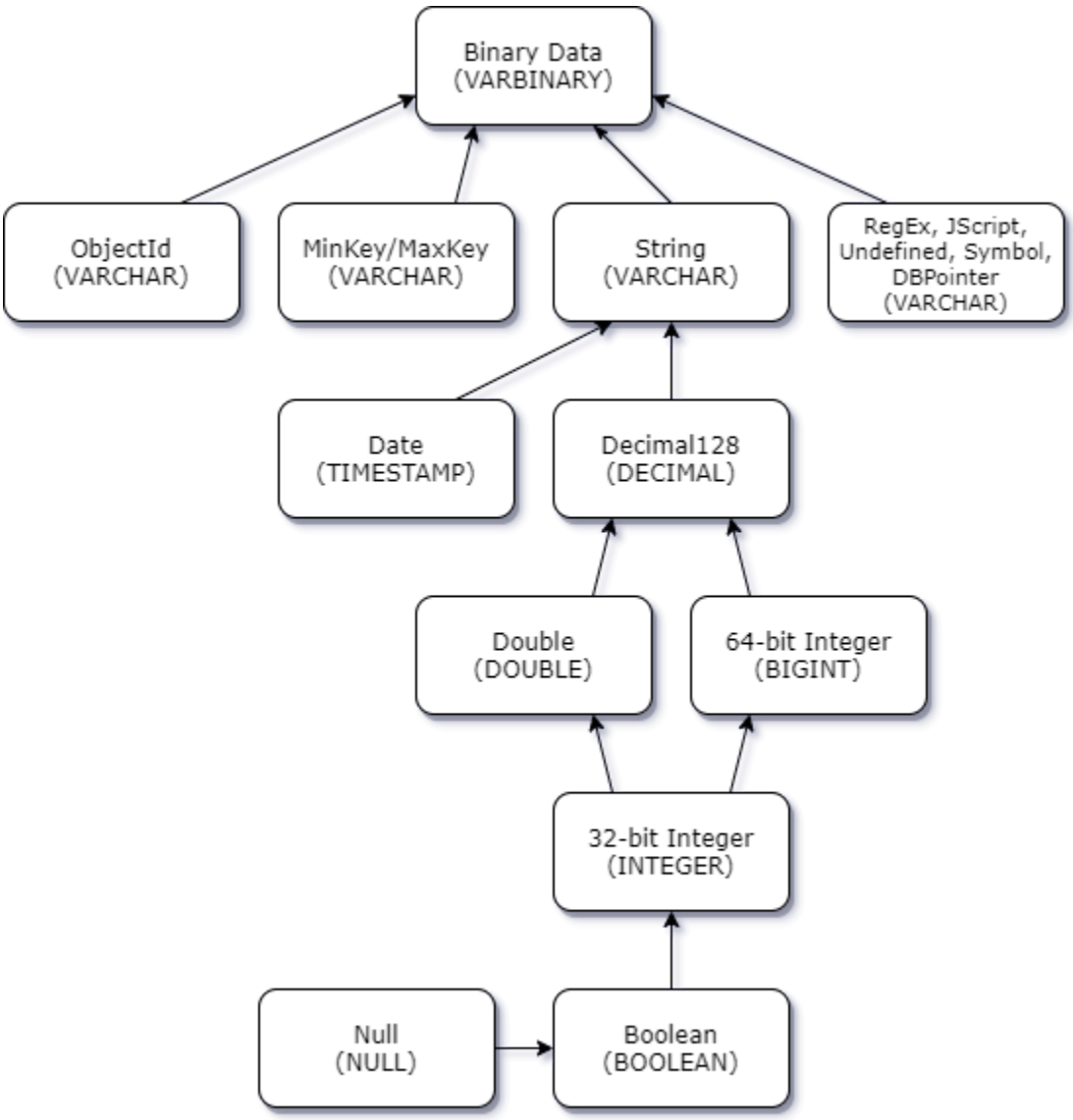
```
"age" : "32"
}
```

Das Altersfeld ist im ersten Dokument vom Typ 32-Bit-Ganzzahl, im zweiten Dokument vom Typ Zeichenfolge. In diesem Fall wird der JDBC-Treiber den JDBC-Datentyp auf VARCHAR heraufstufen, um einen der beiden Datentypen zu verarbeiten, wenn er angetroffen wird.

Tabellenname	Spaltenname	Datentyp	Schlüssel
Mannschaft	Team-ID	VARCHAR	PK
Mannschaft	Name	VARCHAR	
Mannschaft	age	VARCHAR	

Förderung skalar-skalarer Konflikte

Das folgende Diagramm zeigt, wie Konflikte zwischen skalaren und skalaren Datentypen gelöst werden.



Förderung von Konflikten vom Typ skalarkomplex

Wie bei Skalar-Skalar-Typkonflikten kann dasselbe Feld in verschiedenen Dokumenten widersprüchliche Datentypen zwischen komplexen (Array und Objekt) und skalaren Datentypen (Integer, Boolean usw.) aufweisen. All diese Konflikte werden für diese Felder in VARCHAR aufgelöst (heraufgestuft). In diesem Fall werden Array- und Objektdaten als JSON-Darstellung zurückgegeben.

Beispiel für einen Konflikt zwischen eingebettetem Array und Zeichenkettenfeld:

```
{
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
```

```
      "Vogue",
      "People",
      "USA Today"
   ]
}
{
   "_id":"112244",
   "name":"Joan Starr",
   "subscriptions":1
}
```

Das obige Beispiel entspricht dem Schema für die Tabelle customer2:

Tabellenname	Spaltenname	Datentyp	Schlüssel
Kunde2	Kunde2-ID	VARCHAR	PK
Kunde2	Name	VARCHAR	
Kunde2	Abonnement	VARCHAR	

und die virtuelle Tabelle customer1_subscriptions:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer1_subscriptions	Kunde1-ID	VARCHAR	PK/FK
customer1_abonnements	subscriptions_index_lvl0	BIGINT	PK
customer1_abonnements	Wert	VARCHAR	
customer_address	city	VARCHAR	
customer_address	Region	VARCHAR	
customer_address	country	VARCHAR	

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer_address	Code	VARCHAR	

Behandlung von Objekt- und Array-Datentypen

Bisher haben wir nur beschrieben, wie skalare Datentypen zugeordnet werden. Objekt- und Array-Datentypen werden (derzeit) virtuellen Tabellen zugeordnet. Der JDBC-Treiber erstellt eine virtuelle Tabelle, die entweder Objekt- oder Array-Felder in einem Dokument darstellt. Der Name der zugewiesenen virtuellen Tabelle verkettet den Namen der ursprünglichen Sammlung, gefolgt vom Feldnamen, getrennt durch einen Unterstrich („_“).

Der Primärschlüssel der Basistabelle („_id“) nimmt in der neuen virtuellen Tabelle einen neuen Namen an und wird als Fremdschlüssel für die zugehörige Basistabelle bereitgestellt.

Für Felder vom Typ eingebettetes Array werden Indexspalten generiert, um den Index im Array auf jeder Ebene des Arrays darzustellen.

Beispiel für ein eingebettetes Objektfeld

Für Objektfelder in einem Dokument wird vom JDBC-Treiber eine Zuordnung zu einer virtuellen Tabelle erstellt.

```
{
   "Collection: customer",
   "_id":"112233",
   "name":"George Jackson",
   "address":{
      "address1":"123 Avenue Way",
      "address2":"Apt. 5",
      "city":"Hollywood",
      "region":"California",
      "country":"USA",
      "code":"90210"
   }
}
```

Das obige Beispiel ist dem Schema für die Kundentabelle zugeordnet:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer	Kunden-ID	VARCHAR	PK
customer	Name	VARCHAR	

und die virtuelle Tabelle customer_address:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer_address	Kunden-ID	VARCHAR	PK/FK
customer_address	Adresse 1	VARCHAR	
customer_address	Adresse2	VARCHAR	
customer_address	city	VARCHAR	
customer_address	Region	VARCHAR	
customer_address	country	VARCHAR	
customer_address	Code	VARCHAR	

Beispiel für ein eingebettetes Array-Feld

Für Array-Felder in einem Dokument wird vom JDBC-Treiber auch eine Zuordnung zu einer virtuellen Tabelle erstellt.

```
{
   "Collection: customer1",
   "_id":"112233",
   "name":"George Jackson",
   "subscriptions":[
      "Vogue",
      "People",
      "USA Today"
   ]
}
```

Das obige Beispiel ist dem Schema für die Tabelle customer1 zugeordnet:

Tabellenname	Spaltenname	Datentyp	Schlüssel
Kunde1	Kunde1-ID	VARCHAR	PK
Kunde1	Name	VARCHAR	

und die virtuelle Tabelle customer1_subscriptions:

Tabellenname	Spaltenname	Datentyp	Schlüssel
customer1_subscriptions	Kunde1-ID	VARCHAR	PK/FK
customer1_abonnements	subscriptions_index_lvl0	BIGINT	PK
customer1_abonnements	Wert	VARCHAR	
customer_address	city	VARCHAR	
customer_address	Region	VARCHAR	
customer_address	country	VARCHAR	
customer_address	Code	VARCHAR	

SQL-Unterstützung und Einschränkungen

Der Amazon DocumentDB JDBC-Treiber ist ein schreibgeschützter Treiber, der eine Teilmenge von SQL-92 und einige gängige Erweiterungen unterstützt. Weitere Informationen finden Sie in der Dokumentation zu den SQL-Einschränkungen und in der Dokumentation zu den JDBC-Einschränkungen.

Fehlerbehebung

Wenn Sie Probleme mit der Verwendung des Amazon DocumentDB DocumentDB-JDBC-Treibers haben, finden Sie weitere Informationen im Leitfaden zur Fehlerbehebung.

Stellen Sie mithilfe des Amazon DocumentDB DocumentDB-ODBC-Treibers eine Verbindung her

Der ODBC-Treiber für Amazon DocumentDB bietet Entwicklern eine relationale SQL-Schnittstelle und ermöglicht die Konnektivität von BI-Tools wie Power BI Desktop und Microsoft Excel.

Weitere Informationen finden Sie in der Amazon DocumentDB ODBC-Treiber-Dokumentation unter. GitHub

Themen

- Erste Schritte
- Den Amazon DocumentDB ODBC-Treiber in Windows einrichten
- Stellen Sie von Microsoft Excel aus eine Connect zu Amazon DocumentDB her
- Stellen Sie von Microsoft Power BI Desktop aus eine Connect zu Amazon DocumentDB her
- Automatische Schemagenerierung
- SQL-Unterstützung und Einschränkungen
- Fehlerbehebung

Erste Schritte

Schritt 1. Amazon DocumentDB-Cluster erstellen

Wenn Sie noch keinen Amazon DocumentDB-Cluster haben, gibt es eine Reihe von Möglichkeiten, um loszulegen.

> **Note**
>
> Amazon DocumentDB ist ein Dienst, der nur für Virtual Private Cloud (VPC) verfügbar ist. Wenn Sie eine Verbindung von einem lokalen Computer außerhalb der VPC des Clusters herstellen, müssen Sie eine SSH-Verbindung zu einer EC2 Amazon-Instance herstellen. Starten Sie in diesem Fall Ihren Cluster mithilfe der Anweisungen unter Connect with EC2.

Weitere Informationen zum SSH-Tunneling und wann Sie es möglicherweise benötigen, finden Sie unter Verwenden eines SSH-Tunnels zum Herstellen einer Connect zu Amazon DocumentDB.

Schritt 2. JRE- oder JDK-Installation

Abhängig von Ihrer BI-Anwendung müssen Sie möglicherweise sicherstellen, dass eine 64-Bit-JRE- oder JDK-Installationsversion 8 oder höher auf Ihrem Computer installiert ist. Sie können das Java SE Runtime Environment 8 hier herunterladen.

Schritt 3. Laden Sie den Amazon DocumentDB ODBC-Treiber herunter

Laden Sie den Amazon DocumentDB ODBC-Treiber hier herunter. Wählen Sie das richtige Installationsprogramm aus (z. B. documentdb-odbc-1.0.0.msi). Folgen Sie der Installationsanleitung.

Schritt 4. Verwenden eines SSH-Tunnels zum Connect zu Amazon DocumentDB

Amazon DocumentDB-Cluster werden in einer Amazon Virtual Private Cloud (Amazon VPC) bereitgestellt. Sie können direkt von EC2 Amazon-Instances oder anderen AWS Services abgerufen werden, die in derselben Amazon VPC bereitgestellt werden. Darüber hinaus kann Amazon DocumentDB von EC2 Amazon-Instances oder anderen AWS Services VPCs in verschiedenen Regionen derselben AWS Region oder anderen Regionen über VPC-Peering aufgerufen werden.

Nehmen wir jedoch an, dass Ihr Anwendungsfall erfordert, dass Sie (oder Ihre Anwendung) von außerhalb der VPC des Clusters auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zugreifen. Dies wird bei den meisten Benutzern der Fall sein, die ihre Anwendung nicht auf einer VM in derselben VPC wie der Amazon DocumentDB-Cluster ausführen. Wenn Sie eine Verbindung von außerhalb der VPC herstellen, können Sie SSH-Tunneling (auch als Portweiterleitung bezeichnet) verwenden, um auf Ihre Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen.

Um einen SSH-Tunnel zu erstellen, benötigen Sie eine EC2 Amazon-Instance, die in derselben Amazon VPC wie Ihr Amazon DocumentDB-Cluster läuft. Sie können entweder eine bestehende EC2 Instanz in derselben VPC wie Ihr Cluster verwenden oder eine erstellen. Sie können einen SSH-Tunnel zum Amazon DocumentDB-Cluster einrichten, `sample-cluster.node.us-east-1.docdb.amazonaws.com` indem Sie den folgenden Befehl auf Ihrem lokalen Computer ausführen:

```
ssh -i "ec2Access.pem" -L 27017:sample-cluster.node.us-
east-1.docdb.amazonaws.com:27017 ubuntu@ec2-34-229-221-164.compute-1.amazonaws.com -
N
```

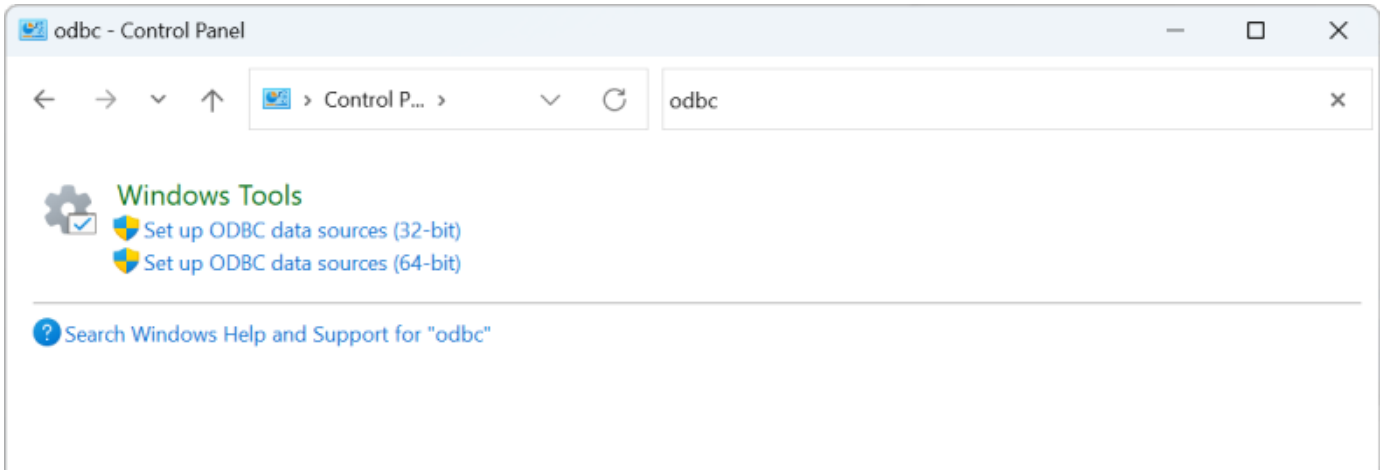
Das `-L`-Flag dient zur Weiterleitung eines lokalen Ports. Dies ist eine Voraussetzung für die Verbindung zu einem BI-Tool, das auf einem Client außerhalb Ihrer VPC ausgeführt wird. Sobald Sie den obigen Schritt ausgeführt haben, können Sie mit den nächsten Schritten für das BI-Tool Ihrer Wahl fortfahren.

Weitere Informationen zum SSH-Tunneling finden Sie in der Dokumentation Using an SSH-Tunnel to Connect to Amazon DocumentDB.

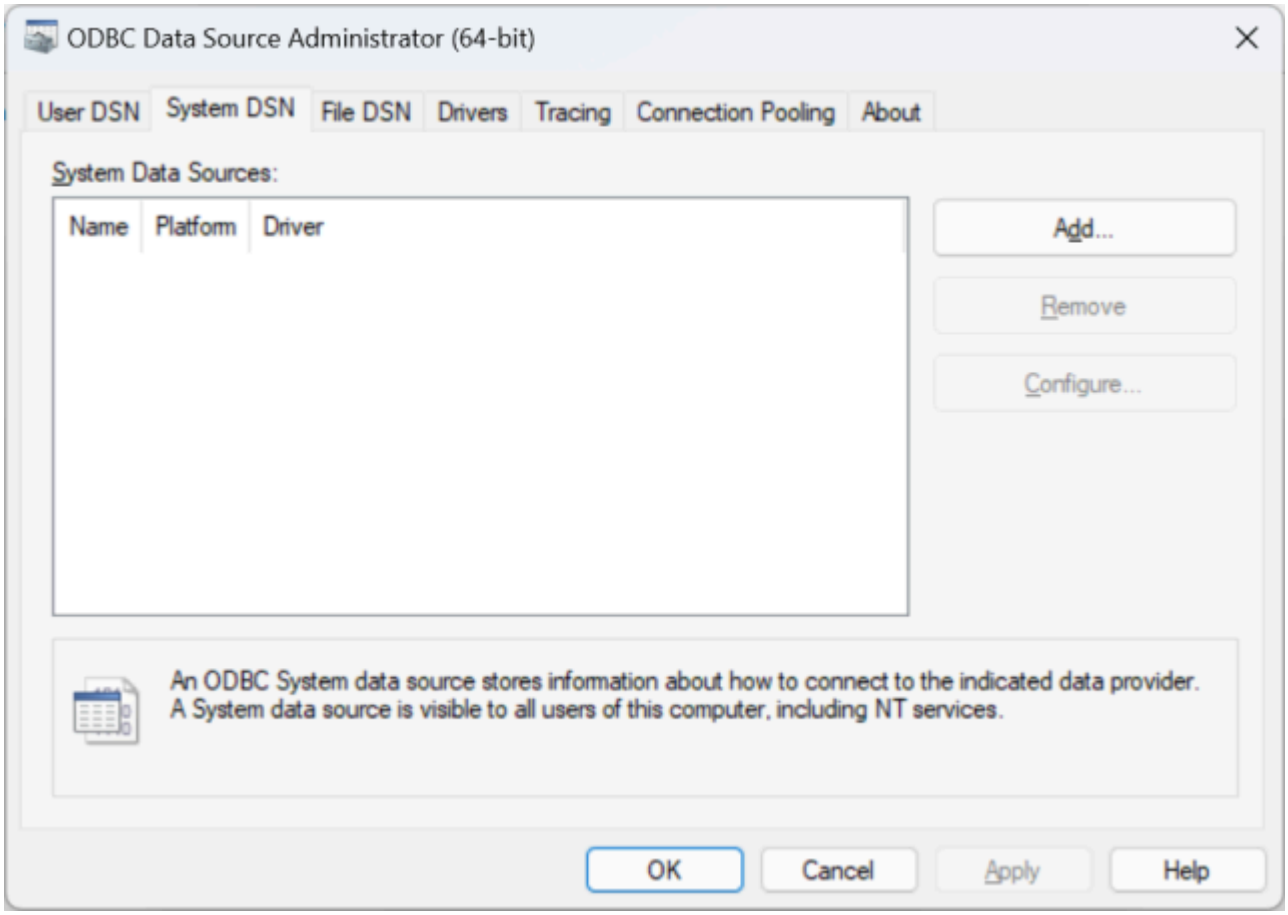
Den Amazon DocumentDB ODBC-Treiber in Windows einrichten

Gehen Sie wie folgt vor, um den Amazon DocumentDB DocumentDB-ODBC-Treiber in Windows einzurichten:

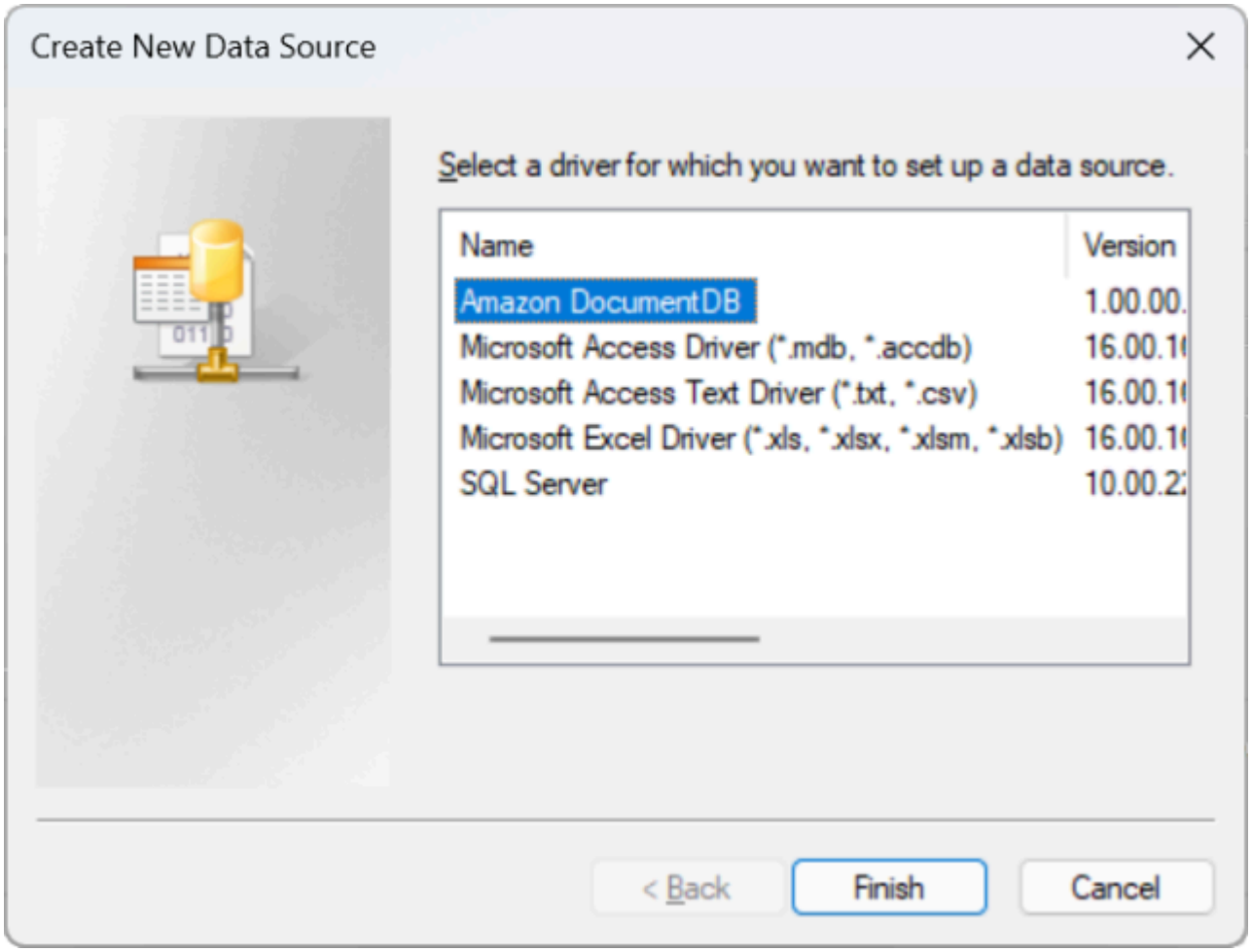
1. Öffnen Sie die Systemsteuerung in Windows und suchen Sie nach ODBC (oder wählen Sie im Menü Windows-Tools > ODBC-Datenquellen (32-Bit) oder ODBC-Datenquellen (64-Bit)):



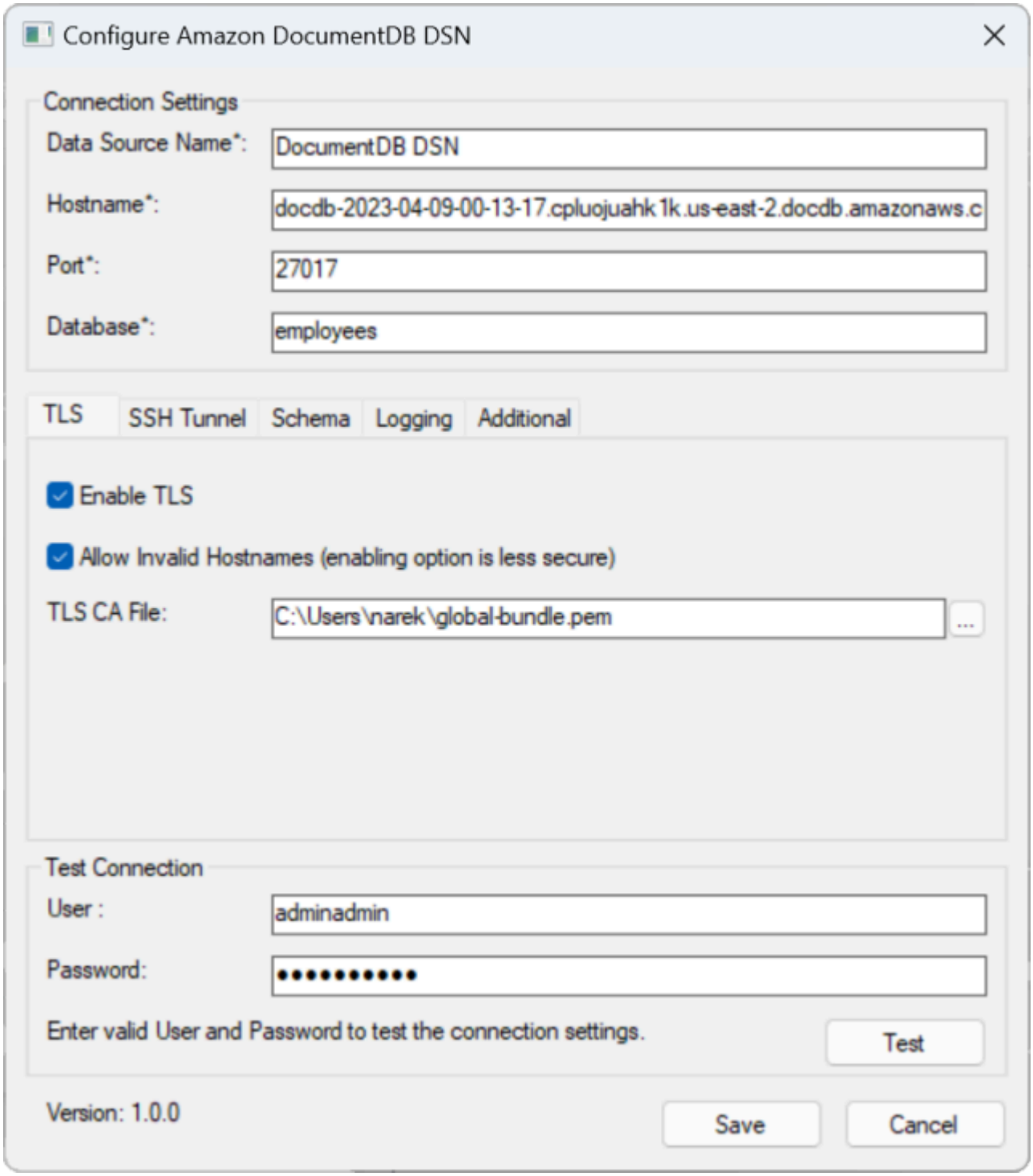
2. Wählen Sie den entsprechenden ODBC-Treiber-Datenquellenadministrator aus: Wählen Sie die 32-Bit-Version, falls diese installiert ist, andernfalls wählen Sie die 64-Bit-Version.
3. Wählen Sie die Registerkarte System-DSN und klicken Sie dann auf Hinzufügen... um einen neuen DSN hinzuzufügen:



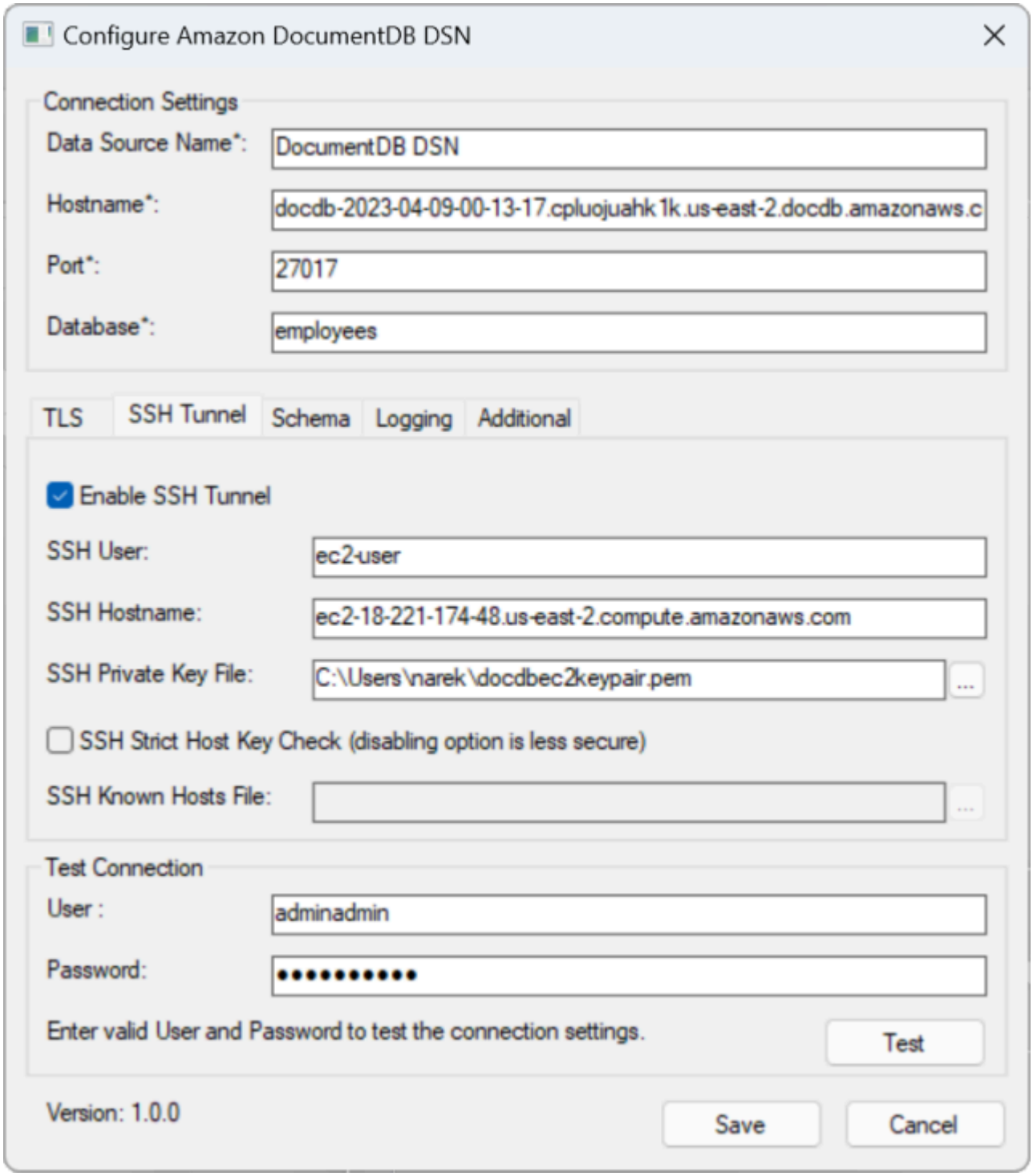
4. Wählen Sie Amazon DocumentDB aus der Liste der Datenquellentreiber aus:



5. Füllen Sie im Dialogfeld Amazon DocumentDB DSN konfigurieren die Felder Verbindungseinstellungen, Registerkarte TLS und Verbindung testen aus und klicken Sie dann auf Speichern:



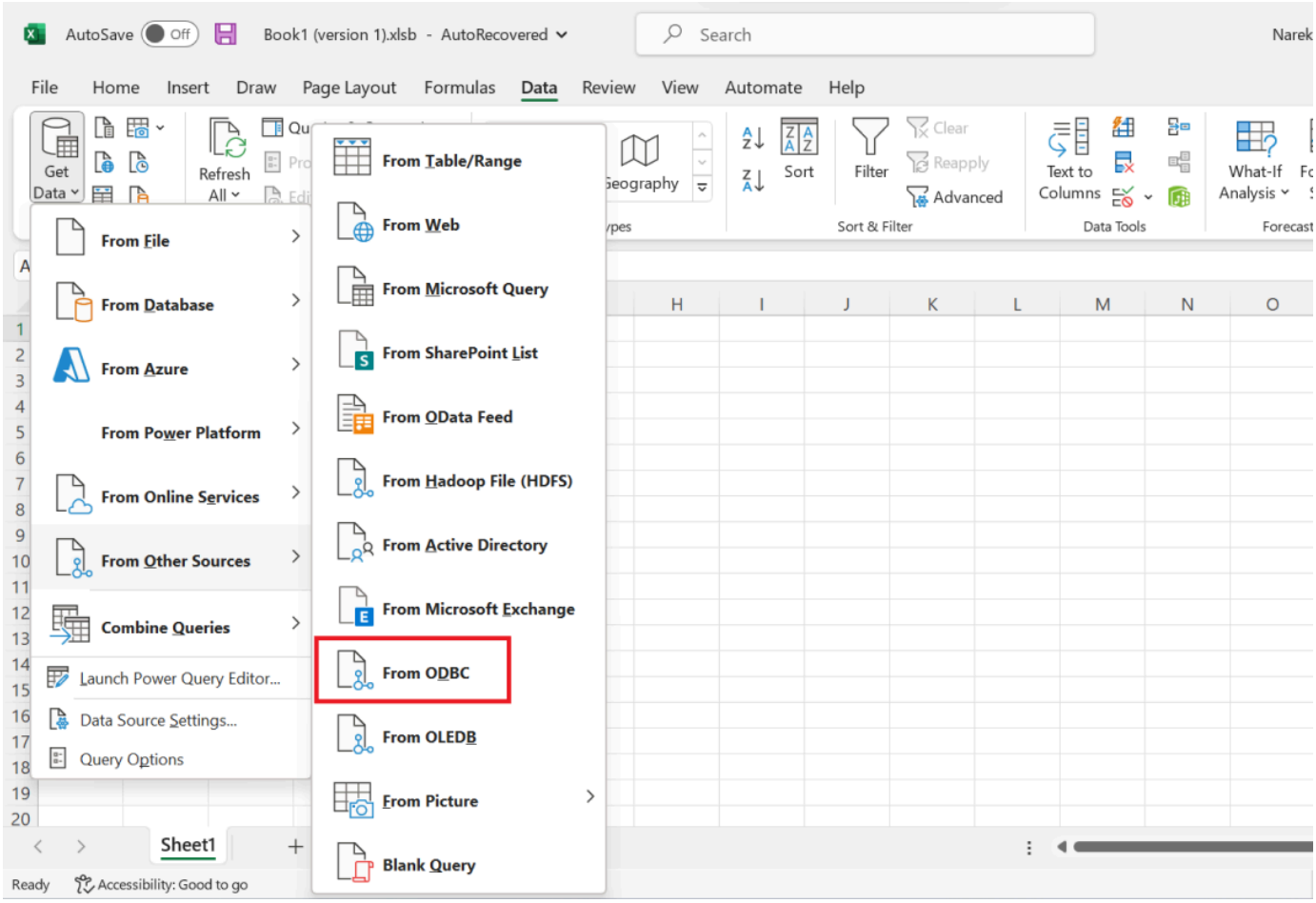
6. Stellen Sie sicher, dass Sie das Windows-Formular korrekt ausfüllen, da die Verbindungsdetails je nach der von Ihnen gewählten SSH-Tunneling-Methode zur Instanz unterschiedlich sein können. EC2 Informationen zu SSH-Tunneling-Methoden finden Sie hier. Weitere Informationen zu den einzelnen Eigenschaften finden Sie unter Syntax und Optionen für Verbindungszeichenfolgen.



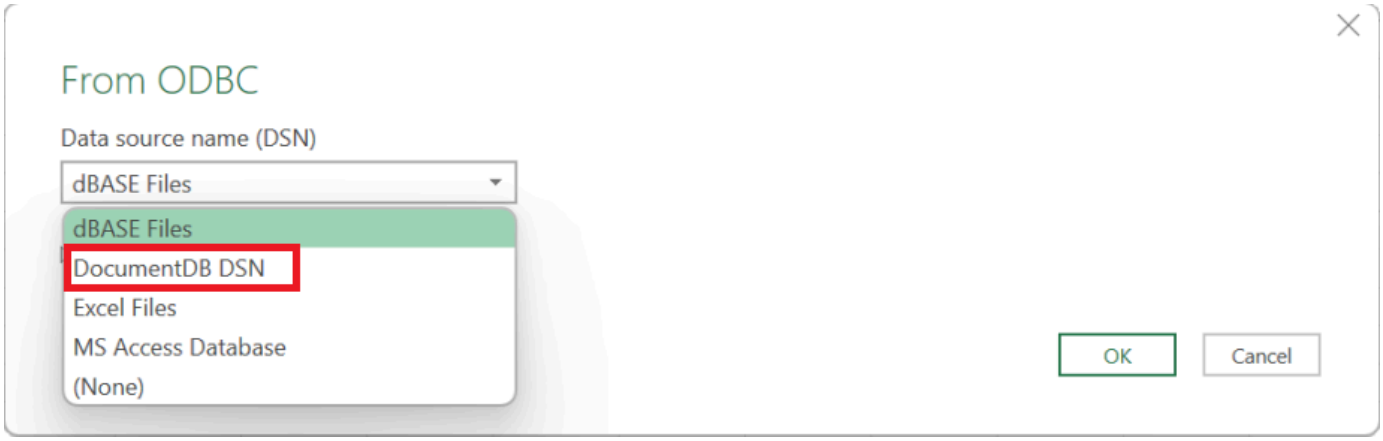
Weitere Informationen zur Konfiguration des Amazon DocumentDB DocumentDB-ODBC-Treibers unter Windows finden Sie hier.

Stellen Sie von Microsoft Excel aus eine Connect zu Amazon DocumentDB her

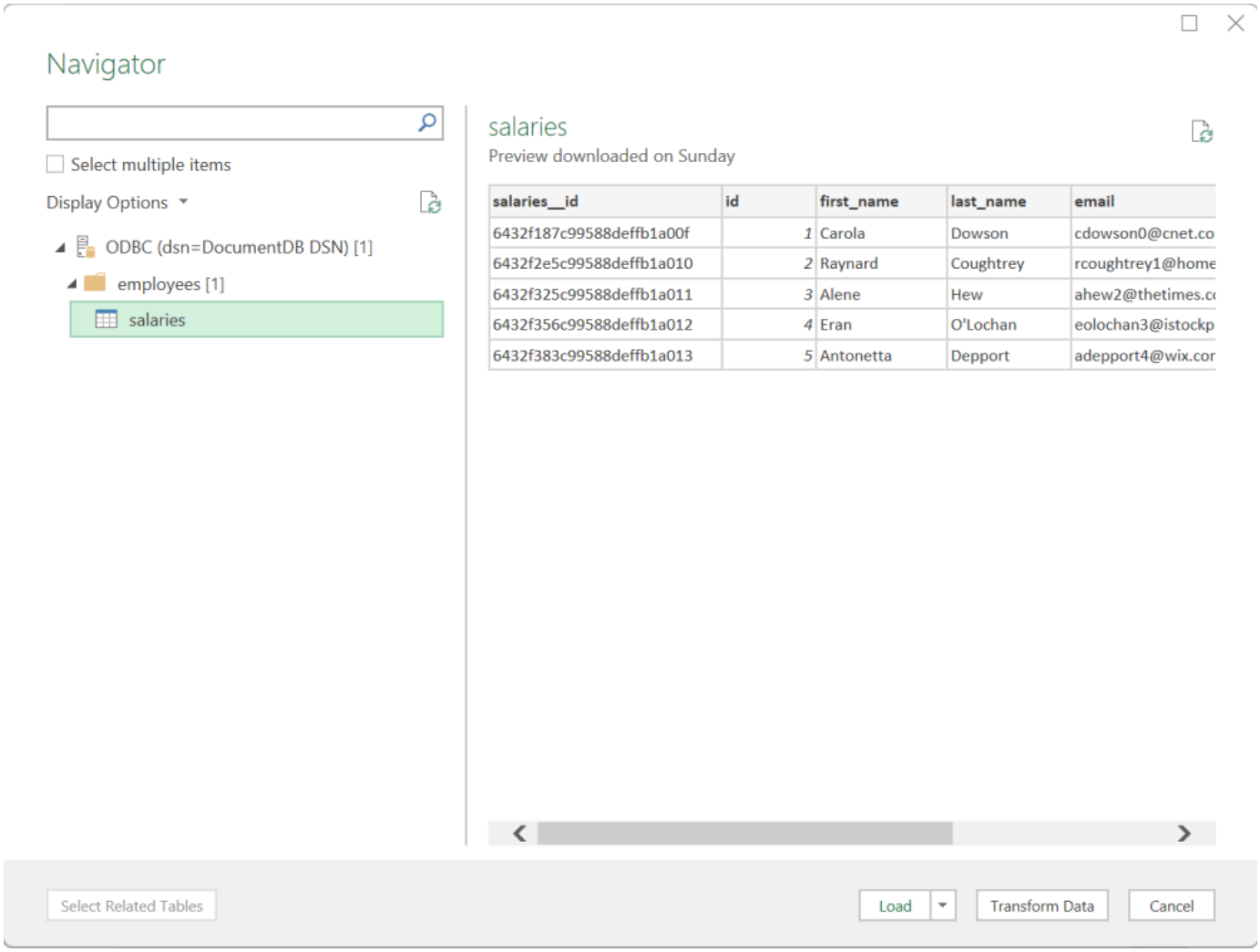
1. Stellen Sie sicher, dass der Amazon DocumentDB-Treiber korrekt installiert und konfiguriert wurde. Weitere Informationen finden Sie unter Einrichten des ODBC-Treibers in Windows.
2. Starten Sie Microsoft Excel.
3. Navigieren Sie zu Daten > Daten abrufen > Aus anderen Quellen.
4. Wählen Sie „Aus ODBC“:



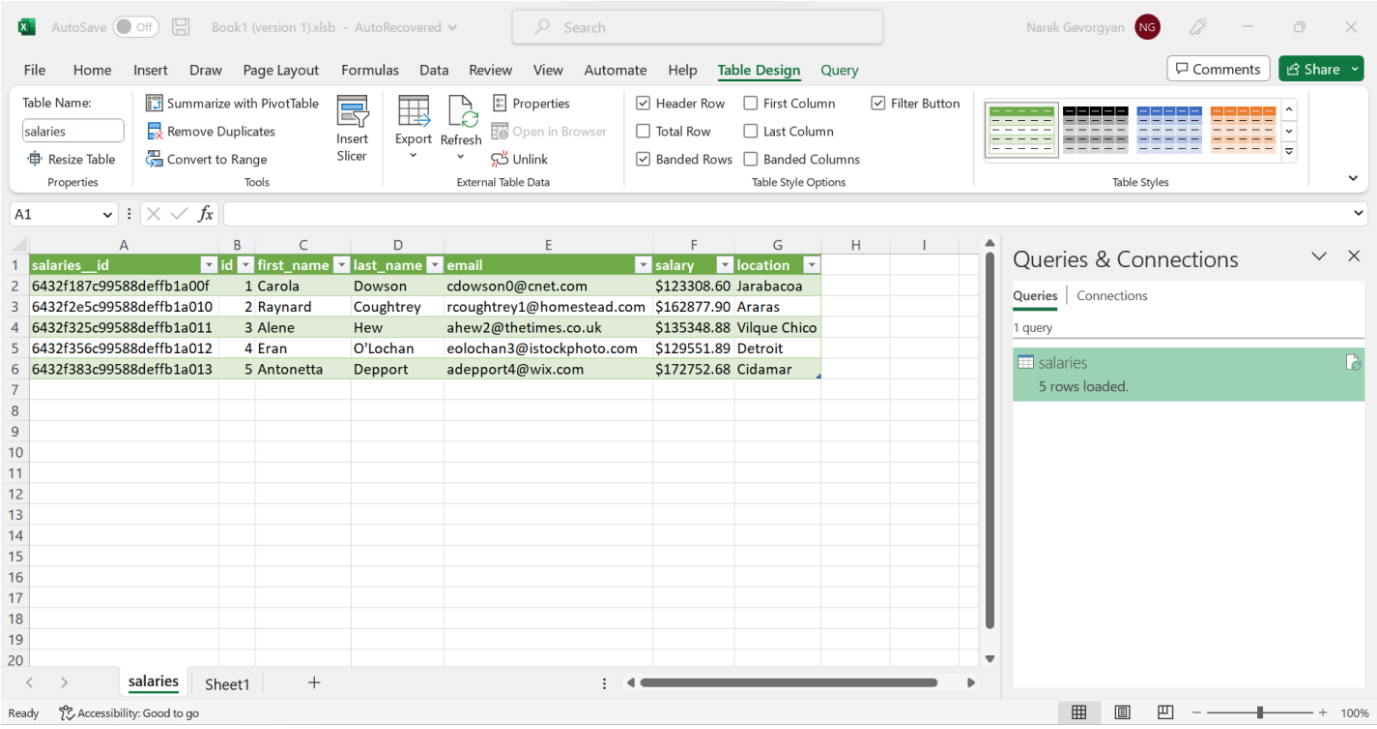
5. Wählen Sie im Dropdownmenü Datenquellenname (DSN) die Datenquelle aus, die Amazon DocumentDB zugeordnet ist:



6. Wählen Sie die Sammlung aus, aus der Sie Daten in Excel laden möchten:



7. Daten in Excel laden:



Stellen Sie von Microsoft Power BI Desktop aus eine Connect zu Amazon DocumentDB her

Themen

- Voraussetzungen
- Benutzerdefinierter Microsoft Power BI Desktop-Connector hinzufügen
- Herstellen einer Verbindung mit dem benutzerdefinierten Amazon DocumentDB DocumentDB-Connector
- Konfiguration von Microsoft Power BI Gateway

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass der Amazon DocumentDB ODBC-Treiber korrekt installiert ist.

Benutzerdefinierter Microsoft Power BI Desktop-Connector hinzufügen

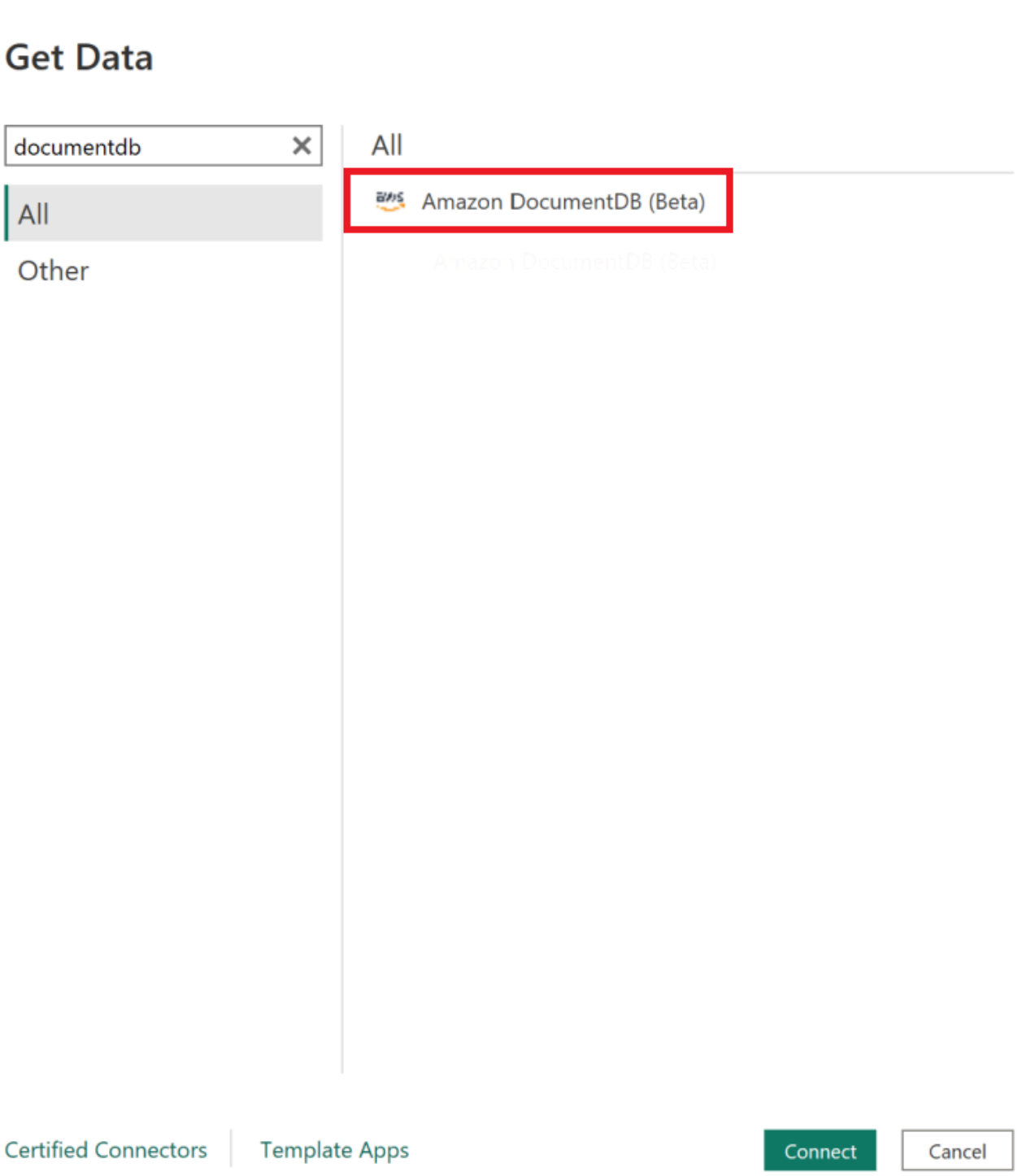
Kopieren Sie die `AmazonDocumentDBConnector.mez` Datei in den `<User>\Documents \Power BI Desktop\Custom Connectors\` Ordner (oder in den Ordner, `<User>\OneDrive`

`\Documents\Power BI Desktop\Custom Connectors` falls Sie ihn verwenden OneDrive). Dadurch kann Power BI auf den benutzerdefinierten Connector zugreifen. Den Connector für Power BI Desktop erhalten Sie hier. Starten Sie Power BI Desktop neu, um sicherzustellen, dass der Connector geladen ist.

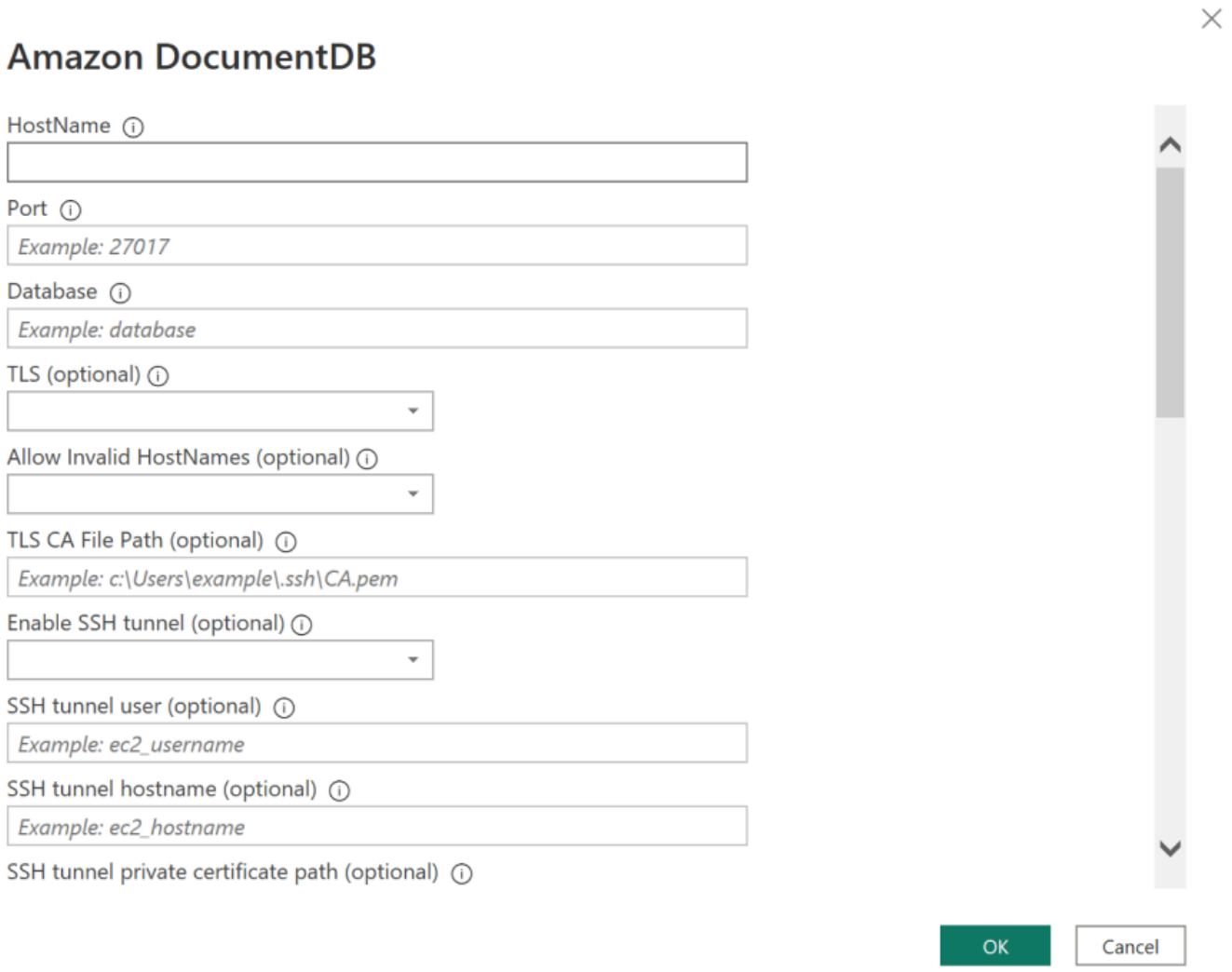
> **Note**
>
> Der benutzerdefinierte Connector unterstützt nur Amazon DocumentDB DocumentDB-Benutzernamen und -Passwort für die Authentifizierung.

Herstellen einer Verbindung mit dem benutzerdefinierten Amazon DocumentDB DocumentDB-Connector

1. Wählen Sie Amazon DocumentDB (Beta) unter Daten abrufen aus und klicken Sie auf Connect. Wenn Sie wegen der Nutzung eines Drittanbieterdienstes eine Warnung erhalten, klicken Sie auf Weiter.



2. Geben Sie alle erforderlichen Informationen ein, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herzustellen, und klicken Sie dann auf OK:

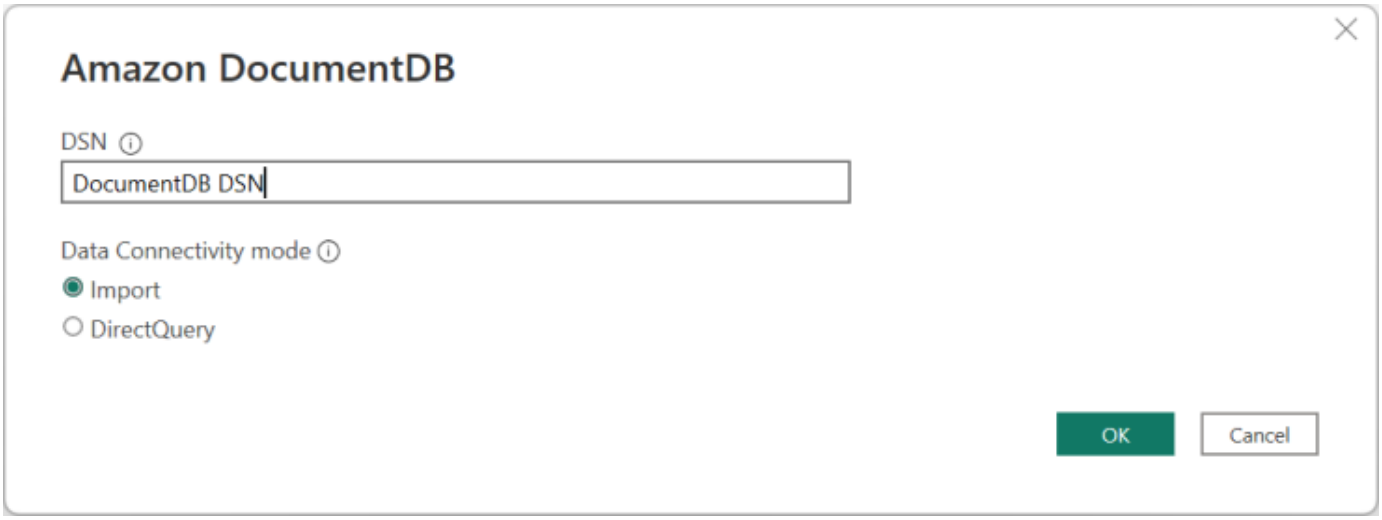


> **Note**
>
> Abhängig von der Konfiguration des Datenquellennamens (DSN) Ihres ODBC-Treibers wird der Bildschirm mit den SSH-Verbindungsdetails möglicherweise nicht angezeigt, wenn Sie die erforderlichen Informationen bereits in den DSN-Einstellungen angegeben haben.

3. Wählen Sie den Datenverbindungsmodus:

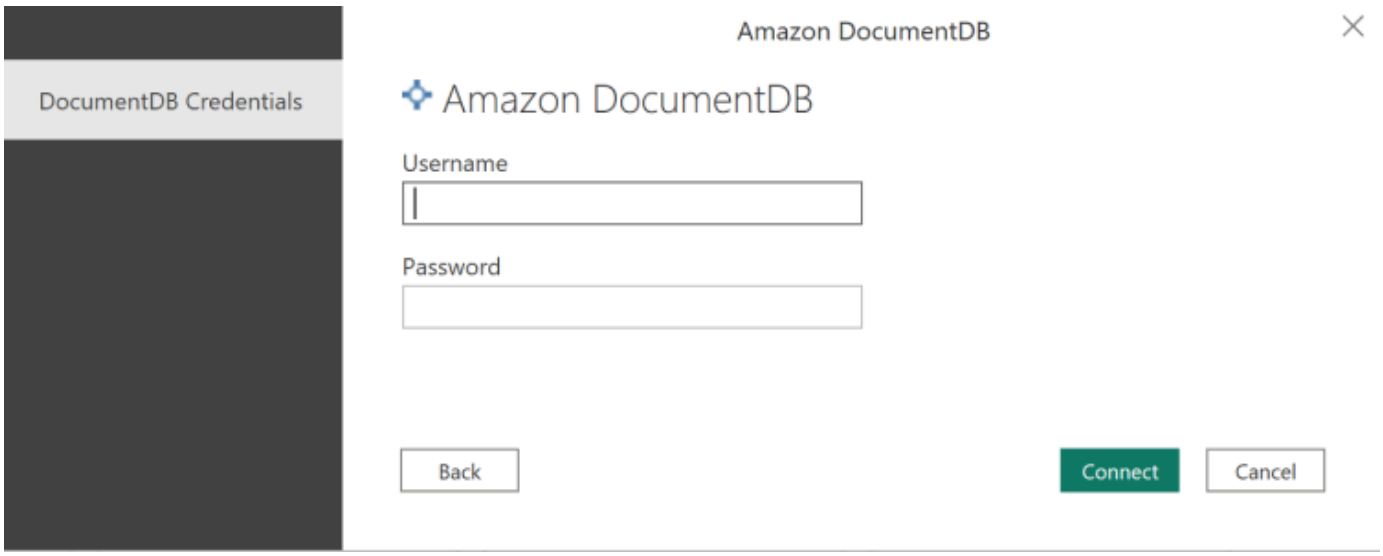
   - Import - lädt alle Daten und speichert die Informationen auf der Festplatte. Die Daten müssen aktualisiert und neu geladen werden, damit Datenaktualisierungen angezeigt werden.

- Direkte Abfrage — lädt keine Daten, sondern führt Live-Abfragen der Daten durch. Das bedeutet, dass Daten nicht aktualisiert und neu geladen werden müssen, um Datenaktualisierungen anzuzeigen.

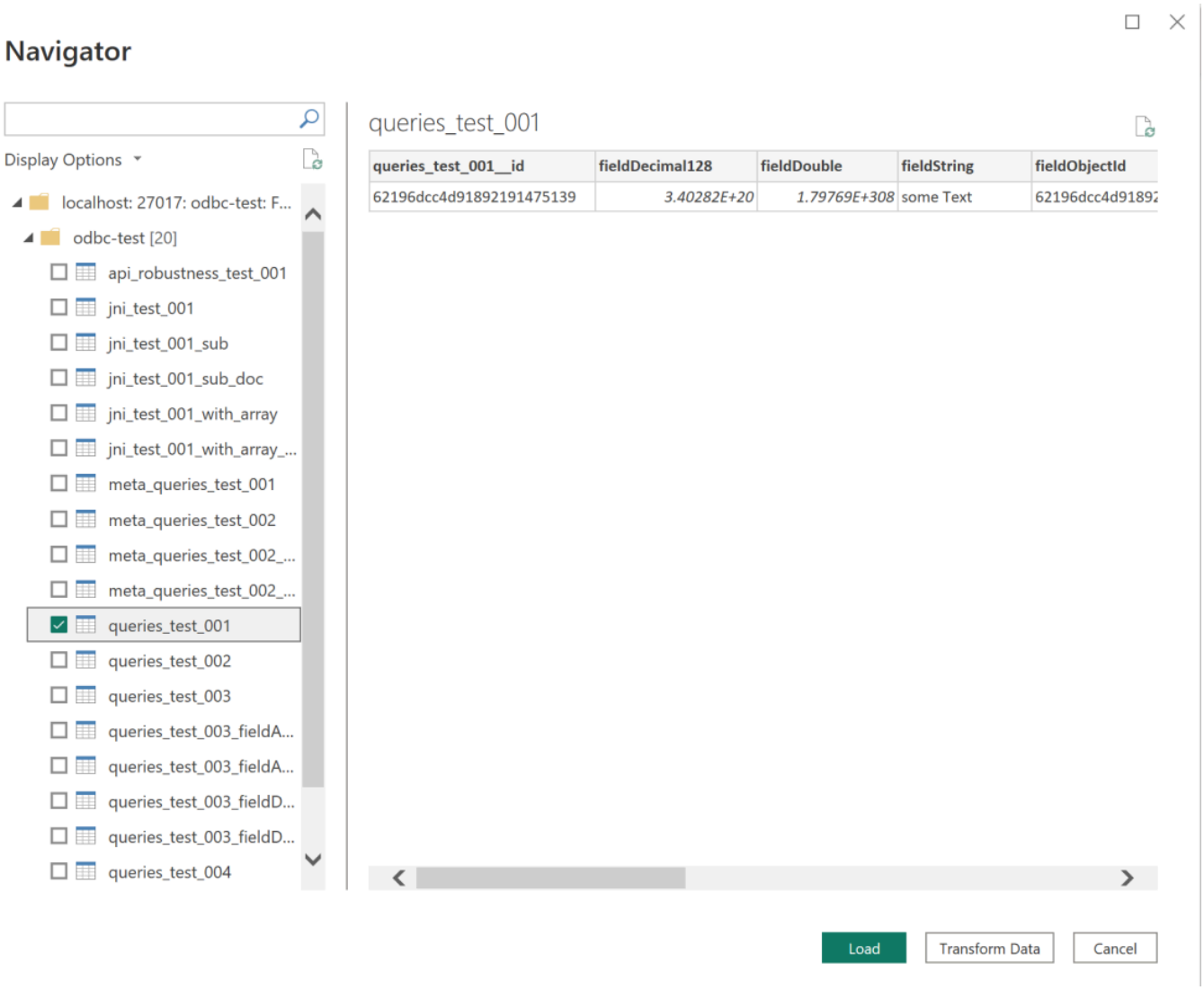


> **Note**
>
> Wenn Sie eine sehr große Datenmenge verwenden, kann das Importieren aller Daten einen längeren Zeitraum in Anspruch nehmen.

4. Wenn Sie zum ersten Mal eine Verbindung zu dieser Datenquelle herstellen, wählen Sie den Authentifizierungstyp aus und geben Sie Ihre Anmeldeinformationen ein, wenn Sie dazu aufgefordert werden. Klicken Sie dann auf Connect:



5. Wählen Sie im Navigator-Dialogfeld die gewünschten Datenbanktabellen aus und klicken Sie dann entweder auf Laden, um die Daten zu laden, oder auf Daten transformieren, um mit der Transformation der Daten fortzufahren.



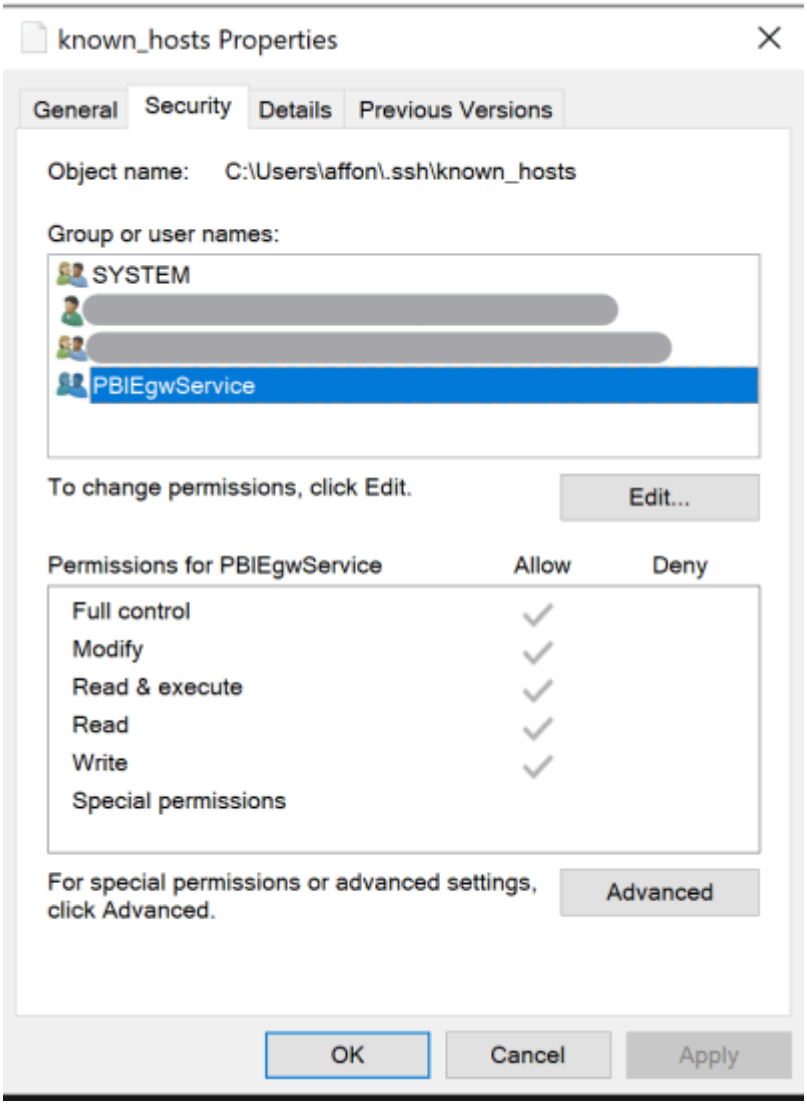
> **Note**
>
> Ihre Datenquelleneinstellungen werden gespeichert, sobald Sie eine Verbindung hergestellt haben. Um sie zu ändern, wählen Sie Daten transformieren > Datenquelleneinstellungen.

Konfiguration von Microsoft Power BI Gateway

Voraussetzungen:

- Stellen Sie sicher, dass der benutzerdefinierte Connector mit Power BI Gateway funktioniert.
- Stellen Sie sicher, dass der ODBC-DSN in den ODBC-Datenquellen auf der Registerkarte System auf dem Computer erstellt wurde, auf dem Power BI Gateway installiert ist.

Wenn Sie die interne SSH-Tunnelfunktion verwenden, `known_hosts` muss sich die Datei dort befinden, wo das Power BI-Dienstkonto Zugriff darauf hat.



> **Note**
>
> Dies gilt auch für alle Dateien, die Sie möglicherweise benötigen, um eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen zu können, wie z. B. eine Zertifizierungsstelle (CA) -Zertifikatsdatei (PEM-Datei).

Automatische Schemagenerierung

Der ODBC-Treiber verwendet den Amazon DocumentDB DocumentDB-JDBC-Treiber über JNI (Java Native Interface), sodass die automatische Schemagenerierung im JDBC-Treiber ähnlich funktioniert. Weitere Informationen zur automatischen Schemagenerierung finden Sie unter Automatische JDBC-Schemagenerierung. Wenn Sie außerdem mehr über die ODBC-Treiberarchitektur erfahren möchten, klicken Sie hier.

SQL-Unterstützung und Einschränkungen

Der Amazon DocumentDB ODBC-Treiber ist ein schreibgeschützter Treiber, der eine Teilmenge von SQL-92 und einige gängige Erweiterungen unterstützt. Weitere Informationen finden Sie in der Dokumentation zu ODBC-Support und Einschränkungen.

Fehlerbehebung

Wenn Sie Probleme mit der Verwendung des Amazon DocumentDB DocumentDB-ODBC-Treibers haben, finden Sie weitere Informationen im Leitfaden zur Fehlerbehebung.

Programm mit Amazon DocumentDB

Der Dienst unterstützt die JSON-Schemavalidierung.

Themen

- Amazon DocumentDB Java-Programmierhandbuch
- Verwenden der JSON-Schemavalidierung

Amazon DocumentDB Java-Programmierhandbuch

Dieses umfassende Handbuch bietet eine detaillierte Anleitung zur Arbeit mit Amazon DocumentDB unter Verwendung der Java-Treiber von MongoDB und behandelt wichtige Aspekte des Datenbankbetriebs und der Datenbankverwaltung.

Themen

- Einführung
- Voraussetzungen

- Datenmodelle für diesen Leitfaden
- Mit einem MongoDB-Java-Treiber eine Verbindung zu Amazon DocumentDB herstellen
- Durchführen von CRUD-Vorgängen in Amazon DocumentDB mit Java
- Indexverwaltung in Amazon DocumentDB mit Java
- Ereignisgesteuerte Programmierung mit Amazon DocumentDB und Java

Einführung

Das Handbuch beginnt mit der Konnektivität und erklärt, wie sichere Verbindungen zu DocumentDB-Clustern mithilfe des MongoDB-Java-Treibers hergestellt werden. Es beschreibt die Komponenten der Verbindungszeichenfolge, die SSL/TLS Implementierung und die verschiedenen Verbindungsoptionen, einschließlich IAM-Authentifizierung und Verbindungspooling, sowie robuste Strategien zur Fehlerbehandlung.

Im Abschnitt CRUD-Operationen (Create, Read, Update, Delete) behandelt das Handbuch ausführlich die Bearbeitung von Dokumenten und zeigt, wie Dokumente sowohl mit Einzel- als auch mit Massenoperationen erstellt, gelesen, aktualisiert und gelöscht werden. Es erklärt die Verwendung von Filtern, Abfragen und verschiedenen Operationsoptionen und betont gleichzeitig bewährte Methoden für die Fehlerbehandlung und die Implementierung von Wiederholungslogik zur Erhöhung der Zuverlässigkeit. Der Leitfaden befasst sich auch ausführlich mit der Indexverwaltung und beschreibt detailliert die Erstellung und Verwaltung verschiedener Indextypen, darunter Einzelfeld-, zusammengesetzte Indizes, Sparse-Indizes und Textindizes. Es wird erklärt, wie die Abfrageleistung durch die richtige Indexauswahl und die Verwendung der `explain()` Funktion zur Analyse von Abfrageausführungsplänen optimiert werden kann.

Der letzte Abschnitt konzentriert sich auf ereignisgesteuertes Programmieren mithilfe der Change-Streams von Amazon DocumentDB und zeigt, wie die Überwachung von Datenänderungen in Echtzeit in Java-Anwendungen implementiert werden kann. Er behandelt die Implementierung von Change-Stream-Cursors, den Umgang mit Resume-Token für den kontinuierlichen Betrieb und zeitbasierte Operationen für die Verarbeitung historischer Daten. Der gesamte Leitfaden enthält praktische Codebeispiele und bewährte Methoden, was ihn zu einer unschätzbaren Ressource für Sie macht, wenn Sie robuste Java-Anwendungen mit Amazon DocumentDB erstellen möchten.

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein AWS Konto mit einem konfigurierten DocumentDB-Cluster. Informationen zur Einrichtung eines DocumentDB-Clusters finden Sie in diesem Blogbeitrag „Erste Schritte“.
- Java Development Kit (JDK) installiert (wir werden Amazon Corretto 21 für dieses Handbuch verwenden).
- Maven für das Abhängigkeitsmanagement.

Datenmodelle für diesen Leitfaden

Der gesamte Beispielcode in diesem Handbuch geht von einer Verbindung zu einer Testdatenbank „ProgGuideData“ aus, die eine Sammlung „Restaurants“ enthält. Alle Beispielcodes in diesem Handbuch funktionieren auf einem Restauranteintragssystem. Im Folgenden finden Sie ein Beispiel dafür, wie ein Dokument in diesem System aussieht:

```
{
    "_id": "ab6ad8f119b5bca3efa2c7ae",
    "restaurantId": "REST-CRT9BL",
    "name": "Thai Curry Palace",
    "description": "Amazing Restaurant, must visit",
    "cuisine": "Thai",
    "address": {
        "street": "914 Park Street",
        "city": "Bryan",
        "state": "AL",
        "zipCode": "96865",
        "location": {
            "type": "Point",
            "coordinates": [-25.4619, 8.389]
        }
    },
    "contact": {
        "phone": "(669) 915-9056 x6657"
    },
    "rating": {
        "average": 3.4,
        "totalReviews": 275
    },
    "priceRange": "$",
    "menu": [{
        "category": "Appetizers",
        "items": [{
            "name": "Buffalo Chicken Wings",
```

```
            "price": 13.42
        }]
    }],
    "features": [
        "Private Dining"
    ],
    "isActive": false“ michelin”: {“
        star”: 3,
        “ranking_years”: 4
    }
}
```

Alle Codebeispiele, die CRUD, Indexverwaltung und ereignisgesteuerte Programmierung zeigen, gehen davon aus, dass Sie über ein `MongoClient`Objekt`db`Client, ein Objekt und ein `MongoDatabase`Objekt `connection`DB verfügen. `MongoCollection`collection

> **Note**
>
> Alle Codebeispiele in diesem Handbuch wurden mit der MongoDB-Java-Treiberversion 5.3.0 getestet.

Mit einem MongoDB-Java-Treiber eine Verbindung zu Amazon DocumentDB herstellen

Dieser Abschnitt enthält eine step-by-step Anleitung für die Verbindung mit Amazon DocumentDB mithilfe von Java-Treibern. Auf diese Weise können Sie mit der Integration von DocumentDB in Ihre Java-Anwendungen beginnen.

Themen

- Schritt 1: Einrichten des Projekts
- Schritt 2: Erstellen Sie die Verbindungszeichenfolge
- Schritt 3: Schreiben Sie den Verbindungscode
- Schritt 4: Verbindungsausnahmen behandeln
- Schritt 5: Den Code ausführen
- Bewährte Methoden für Verbindungen

Schritt 1: Einrichten des Projekts

1. Erstellen Sie mit Maven ein Java-Projekt:

```
mvn archetype:generate -DgroupId=com.docdb.guide -DartifactId=my-docdb-project -
DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

2. Fügen Sie den MongoDB-Java-Treiber als Abhängigkeit für das Projekt in Ihrer Datei 'pom.xml' hinzu:

```
<dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongodb-driver-sync</artifactId>
    <version>5.3.0</version>
</dependency>
```

Schritt 2: Erstellen Sie die Verbindungszeichenfolge

Die Amazon DocumentDB DocumentDB-Verbindungszeichenfolge ist wichtig, um eine Verbindung zwischen Ihrer Anwendung und Ihrem DocumentDB-Cluster herzustellen. Diese Zeichenfolge enthält wichtige Informationen wie den Cluster-Endpunkt, den Port, Authentifizierungsdetails und verschiedene Verbindungsoptionen. Um eine DocumentDB-Verbindungszeichenfolge zu erstellen, beginnen Sie normalerweise mit dem Basisformat:

```
"mongodb://username:password@cluster-endpoint:port/?[connection options]"
```

Sie müssen „Benutzername“ und „Passwort“ durch Ihre tatsächlichen Anmeldeinformationen ersetzen. Den Endpunkt und die Portnummer Ihres Clusters finden Sie sowohl im AWS Management Console als auch über AWS CLI. Informationen Die Endpunkte eines Clusters finden zum Finden des Cluster-Endpunkts für Ihren Cluster finden Sie unter. Der Standardport für DocumentDB ist 27017.

Beispiele für Verbindungszeichenfolgen

- Herstellen einer Verbindung zu DocumentDB mithilfe von Verschlüsselung bei der Übertragung und Sicherstellung, dass Leseanforderungen an Read Replicas und Schreiben an Primary weitergeleitet werden:

```
"mongodb://username:password@cluster-endpoint:27017/?tls=true&
   tlsCAFile=global-bundle.pem&
   readPreference=secondaryPreferred&
```

```
   retryWrites=false"
```

- Herstellen einer Verbindung zu DocumentDB mithilfe der IAM-Authentifizierung:

```
"mongodb://cluster-endpoint:27017/?tls=true&
   tlsCAFile=global-bundle.pem&
   readPreference=secondaryPreferred&
   retryWrites=false&
   authSource=%24external&
   authMechanism=MONGODB-AWS"
```

Die verschiedenen Optionen, die für die Verbindungszeichenfolge verfügbar sind, lauten wie folgt:

- TLS-Zertifikat
- Aus gelesenen Replikaten lesen
- Schreiben Sie ein Anliegen und führen Sie ein Tagebuch
- RetryWrites
- IAM-Authentifizierung
- Verbindungspool
- Parameter für das Verbindungs-Timeout

TLS-Zertifikat

`tls=true|false`— Diese Option aktiviert oder deaktiviert Transport Layer Security (TLS). Standardmäßig ist die Verschlüsselung bei der Übertragung auf dem Amazon DocumentDB-Cluster aktiviert. Daher sollte der Wert für diese Option lauten, sofern TLS nicht auf Cluster-Ebene deaktiviert ist. `true`

Bei Verwendung von TLS muss der Code beim Herstellen einer Verbindung zu einem DocumentDB-Cluster ein SSL-Zertifikat bereitstellen. Laden Sie das Zertifikat herunter, das für die sichere Verbindung zum Cluster erforderlich ist: `global-bundle.pem`. Es gibt zwei Möglichkeiten, die `global-bundle.pem` Datei zu verwenden.

- Option 1 — Extrahieren Sie alle Zertifikate aus der `global-bundle.pem` Datei und speichern Sie sie mit dem Keytool von Java in einer `.jks` Datei, die später im Code verwendet werden kann. Das Skript, das zeigt, wie Verbindung bei aktiviertem TLS herstellen das geht, finden Sie auf der Registerkarte Java.

- Option 2 — Fügen Sie die `global-bundle.pem` Datei dynamisch zum Code hinzu, erstellen Sie einen Keystore im Arbeitsspeicher und verwenden Sie ihn`SSLContext`, um das Zertifikat beim Herstellen der Verbindung bereitzustellen.

Aus gelesenen Replikaten lesen

`replicaSet=rs0&readPreference=secondaryPreferred`— Wenn Sie diese beiden Optionen angeben, werden alle Leseanforderungen an die Read Replicas und Schreibanforderungen an die primäre Instanz weitergeleitet. Durch die Verwendung `replicaSet=rs0` in der Verbindungszeichenfolge kann der MongoDB-Treiber eine automatisch aktualisierte Ansicht der Cluster-Topologie beibehalten, sodass Anwendungen die Sichtbarkeit der aktuellen Knotenkonfigurationen beibehalten können, wenn Instanzen hinzugefügt oder entfernt werden. Wenn Sie diese Optionen nicht angeben oder angeben, werden alle Lese- und Schreibvorgänge an die primäre Instanz `readPreference=primary` gesendet. Weitere Optionen für finden `readPreference` Sie unterLesen Sie die Einstellungsoptionen.

Schreiben Sie ein Anliegen und führen Sie ein Tagebuch

Das Problem mit Schreibvorgängen bestimmt den Grad der Bestätigung, die von der Datenbank für Schreibvorgänge angefordert wird. MongoDB-Treiber bieten eine Option zur Optimierung von Schreib-, Bedenken- und Journaldateien. Amazon DocumentDB erwartet nicht, dass Sie Write Concern und Journal festlegen, und ignoriert die Werte, die für `w` und `j` (`writeConcern`und`journal`) gesendet werden. DocumentDB schreibt Daten immer mit`writeConcern`: `majority` und`journal`:, `true` sodass die Schreibvorgänge auf den meisten Knoten dauerhaft aufgezeichnet werden, bevor eine Bestätigung an den Client gesendet wird.

RetryWrites

`retryWrites=false`— DocumentDB unterstützt keine wiederholbaren Schreibvorgänge und daher sollte dieses Attribut immer auf gesetzt werden. `false`

IAM-Authentifizierung

`authSource=%24external`und **`authMechanism=MONGODB-AWS`** — Diese beiden Parameter werden zur Authentifizierung mit verwendet. AWS Identity and Access Management Die IAM-Authentifizierung ist derzeit nur in der instanzbasierten Cluster-Version 5.0 verfügbar. Weitere Informationen finden Sie unter Authentifizierung mit IAM-Identität.

Verbindungspool

Diese Optionen sind für das Verbindungspooling verfügbar:

- **`maxPoolSize`**— Legt die maximale Anzahl von Verbindungen fest, die im Pool erstellt werden können. Wenn alle Verbindungen verwendet werden und eine neue Anfrage eingeht, wartet es darauf, dass eine Verbindung verfügbar wird. Die Standardeinstellung für MongoDB-Java-Treiber ist 100.
- **`minPoolSize`**— Gibt die Mindestanzahl von Verbindungen an, die jederzeit im Pool aufrechterhalten werden sollten. Die Standardeinstellung für MongoDB-Java-Treiber ist 0.
- **`maxIdleTimeMS`**— Legt fest, wie lange eine Verbindung im Pool inaktiv bleiben kann, bevor sie geschlossen und entfernt wird. Die Standardeinstellung für MongoDB-Java-Treiber ist 100 Millisekunden.
- **`waitQueueTimeoutMS`**— Konfiguriert, wie lange ein Thread warten soll, bis eine Verbindung verfügbar wird, wenn der Pool seine maximale Größe erreicht hat. Wenn innerhalb dieser Zeit keine Verbindung verfügbar ist, wird eine Ausnahme ausgelöst. Der Standardwert für MongoDB-Java-Treiber ist 120.000 Millisekunden (2 Minuten).

Parameter für das Verbindungs-Timeout

Timeout ist ein Mechanismus zur Begrenzung der Zeit, die ein Vorgang oder ein Verbindungsversuch dauern kann, bis er als fehlgeschlagen betrachtet wird. Die folgenden Timeout-Parameter sind verfügbar, um unbefristete Wartezeiten zu verhindern und die Ressourcenzuweisung zu verwalten:

- **`connectTimeoutMS`**— Konfiguriert, wie lange der Treiber warten soll, um eine Verbindung zum Cluster herzustellen. Die Standardeinstellung ist 10.000 Millisekunden (10 Sekunden).
- **`socketTimeoutMS`**— Gibt an, wie lange der Treiber bei einem Nicht-Schreibvorgang auf eine Antwort vom Server wartet. Die Standardeinstellung ist 0 (kein Timeout oder unendlich).
- **`serverSelectionTimeoutMS`**— Gibt an, wie lange der Treiber warten wird, um einen verfügbaren Server im Cluster zu finden. Der Standardwert für diese Einstellung ist 30 Sekunden und reicht aus, damit beim Failover eine neue primäre Instanz ausgewählt werden kann.

Schritt 3: Schreiben Sie den Verbindungscode

Das folgende Codebeispiel zeigt, wie eine TLS-Verbindung zu Amazon DocumentDB hergestellt wird:

- Es erstellt die Objekte Java `KeyStore`und `SSLContext>`.

- Es erstellt das MongoClientSettingsObjekt auch, indem es es an das ConnectionStringObjekt übergibt. Um eine TLS-Verbindung herzustellen, müssen Sie das `MongoClientSettings` Objekt verwenden, um das `connectionstring` und zu bind`sslcontext`.
- Using MongoClientsruft ein MongoClientObjekt ab.

```
public static MongoClient makeDbConnection(String dbName, String DbUserName, String
 DbPassword,
    String DbClusterEndPoint, String keyStorePass) throws Exception {
    MongoClient connectedClient;
    String connectionOptions = "?
replicaSet=rs0&readPreference=secondaryPreferred&retryWrites=false";
    String connectionUrl = "mongodb://" + DbUserName + ":" + DbPassword + "@" +
 DbClusterEndPoint + ":27017/" +
        dbName + connectionOptions;

    try {
        KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
        try (FileInputStream fis = new FileInputStream("src/main/resources/certs/
truststore.jks")) {
            trustStore.load(fis, keyStorePass.toCharArray());
            TrustManagerFactory tmf =
 TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
            tmf.init(trustStore);

            SSLContext sslContext = SSLContext.getInstance("TLSv1.2");
            sslContext.init(null, tmf.getTrustManagers(), new SecureRandom());
            ConnectionString connectionString = new ConnectionString(connectionUrl);
            MongoClientSettings settings = MongoClientSettings.builder()
                .applyConnectionString(connectionString)
                .applyToSslSettings(builder - > {
                    builder.enabled(true);
                    builder.context(sslContext);
                })
                .build();
            connectedClient = MongoClients.create(settings);
        }
        return connectedClient;
    } catch (MongoException e5) {
        throw new RuntimeException(e5);
    } catch (Exception e) {
```

```
        throw new RuntimeException(e);
    }
}
```

Schritt 4: Verbindungsausnahmen behandeln

Bei der Arbeit mit DocumentDB in Java-Anwendungen ist die Behandlung von Verbindungsausnahmen entscheidend für die Aufrechterhaltung robuster und zuverlässiger Datenbankoperationen. Durch die richtige Verwaltung helfen diese Ausnahmen nicht nur bei der schnellen Diagnose von Problemen, sondern stellen auch sicher, dass Ihre Anwendung mit vorübergehenden Netzwerkunterbrechungen oder Serverausfällen problemlos umgehen kann, was zu einer verbesserten Stabilität und Benutzererfahrung führt. Zu den wichtigsten Ausnahmen im Zusammenhang mit dem Verbindungsaufbau gehören:

- **`MongoException`**— Eine allgemeine Ausnahme, die für verschiedene Szenarien gelten könnte, die nicht durch speziellere Ausnahmen abgedeckt sind. Stellen Sie sicher, dass diese Ausnahme nach allen anderen spezifischen Ausnahmen behandelt wird, da es sich um eine catch MongoDB-Ausnahme handelt.
- **`MongoTimeoutException`**— Wird ausgegeben, wenn bei einem Vorgang das Zeitlimit überschritten wird. Zum Beispiel die Abfrage eines nicht vorhandenen Cluster-Endpunkts.
- **`MongoSocketException`**— Ausgestellt für Netzwerkprobleme. Zum Beispiel eine plötzliche Netzwerkunterbrechung während eines Vorgangs.
- **`MongoSecurityException`**— Wird ausgegeben, wenn die Authentifizierung fehlschlägt. Zum Beispiel, wenn eine Verbindung mit falschen Anmeldeinformationen hergestellt wird.
- **`MongoConfigurationException`**— Wird ausgegeben, wenn in der Client-Konfiguration ein Fehler auftritt. Zum Beispiel, wenn eine ungültige Verbindungszeichenfolge verwendet wird.

Schritt 5: Den Code ausführen

Das folgende Codebeispiel erstellt eine Amazon DocumentDB DocumentDB-Verbindung und druckt alle Datenbanken:

```
public static void TestConnection() {
    try (MongoClient mongoClient = makeDbConnection(DATABASE_NAME, DB_USER_NAME,
 DB_PASSWORD, DB_CLUSTER_ENDPOINT, KEYSTORE_PASSWORD)) {
        List < String > databases = mongoClient.listDatabaseNames().into(new ArrayList
 < > ());
```

```
        System.out.println("Databases: " + databases);
    } catch (MongoException e) {
        System.err.println("MongoDB error: " + e.getMessage());
        throw new RuntimeException(e);
    }
}
```

Bewährte Methoden für Verbindungen

Im Folgenden finden Sie bewährte Methoden, die Sie berücksichtigen sollten, wenn Sie mit einem MongoDB-Java-Treiber eine Verbindung zu Amazon DocumentDB herstellen:

- Schließen Sie Ihren immer `MongoClient`dann, wenn Sie den Client nicht mehr benötigen, um Ressourcen freizugeben.
- Behandeln Sie Ausnahmen angemessen und implementieren Sie eine korrekte Fehlerprotokollierung.
- Verwenden Sie Umgebungsvariablen oder speichern AWS Secrets Manager Sie vertrauliche Informationen wie Benutzernamen und Passwörter.

Durchführen von CRUD-Vorgängen in Amazon DocumentDB mit Java

In diesem Abschnitt wird die Durchführung von CRUD-Vorgängen (Create, Read, Update, Delete) in Amazon DocumentDB mithilfe von MongoDB-Java-Treibern beschrieben.

Themen

- Dokumente in einer DocumentDB-Sammlung erstellen und einfügen
- Lesen und Abrufen von Daten aus einer DocumentDB-Sammlung
- Aktualisieren vorhandener Dokumente in einer DocumentDB-Sammlung
- Dokumente aus einer DocumentDB-Sammlung entfernen
- Fehlerbehandlung mit Wiederholungslogik

Dokumente in einer DocumentDB-Sammlung erstellen und einfügen

Durch das Einfügen von Dokumenten in Amazon DocumentDB können Sie Ihren Sammlungen neue Daten hinzufügen. Je nach Ihren Bedürfnissen und dem Datenvolumen, mit dem Sie arbeiten, gibt es mehrere Möglichkeiten, Einfügungen durchzuführen. Die einfachste Methode zum Einfügen eines

einzelnen Dokuments in die Sammlung ist. `insertOne()` Um mehrere Dokumente gleichzeitig einzufügen, können Sie die `insertMany()` Methode verwenden, mit der Sie eine Reihe von Dokumenten in einem einzigen Vorgang hinzufügen können. Eine andere Methode zum Einfügen vieler Dokumente in eine DocumentDB-Sammlung ist. `bulkWrite()` In diesem Handbuch werden all diese Methoden zum Erstellen von Dokumenten in einer DocumentDB-Sammlung beschrieben.

`insertOne()`

Lassen Sie uns zunächst untersuchen, wie ein einzelnes Dokument in eine Amazon DocumentDBB-Sammlung eingefügt wird. Das Einfügen eines einzelnen Dokuments erfolgt mithilfe der Methode. `insertOne()` Diese Methode verwendet BsonDocumentfür das Einfügen einen Wert und gibt ein `InsertOneResult`Objekt zurück, mit dem die Objekt-ID des neu eingefügten Dokuments abgerufen werden kann. Der folgende Beispielcode zeigt das Einfügen eines Restaurantdokuments in die Sammlung:

```
Document article = new Document()
    .append("restaurantId", "REST-21G145")
    .append("name", "Future-proofed Intelligent Bronze Hat")
    .append("cuisine", "International")
    .append("rating", new Document()
        .append("average", 1.8)
        .append("totalReviews", 267))
    .append("features", Arrays.asList("Outdoor Seating", "Live Music"));

try {
    InsertOneResult result = collection.insertOne(article);
    System.out.println("Inserted document with the following id: " +
 result.getInsertedId());
} catch (MongoWriteException e) {
    // Handle duplicate key or other write errors
    System.err.println("Failed to insert document: " + e.getMessage());
    throw e;
} catch (MongoException e) {
    // Handle other MongoDB errors
    System.err.println("MongoDB error: " + e.getMessage());
    throw e;
}
```

Achten Sie bei der Verwendung daraufinsertOne(), eine angemessene Fehlerbehandlung einzubeziehen. Im obigen Code hat „restaurantId“ beispielsweise einen eindeutigen Index. Wenn Sie diesen Code erneut ausführen, wird Folgendes ausgelöstMongoWriteException:

```
Failed to insert document: Write operation error on server
 docdbCluster.docdb.amazonaws.com:27017.
Write error: WriteError{code=11000, message='E11000 duplicate key error collection:
 Restaurants index: restaurantId_1', details={}}.
```

insertMany ()

Die wichtigsten Methoden, die zum Einfügen vieler Dokumente in eine Sammlung verwendet werden, sind insertMany () und. `bulkWrite()`

Die `insertMany()` Methode ist die einfachste Methode, mehrere Dokumente in einem einzigen Vorgang einzufügen. Sie akzeptiert eine Liste von Dokumenten und fügt sie in die Sammlung ein. Diese Methode ist ideal, wenn Sie einen Stapel neuer Dokumente einfügen, die unabhängig voneinander sind und keine speziellen Verarbeitungs- oder Mischoperationen erfordern. Der folgende Code zeigt, wie JSON-Dokumente aus einer Datei gelesen und in die Sammlung eingefügt werden. Die `insertMany()` Funktion gibt ein `InsertManyResult`InsertManyResultObjekt zurück, mit dem IDs alle eingefügten Dokumente abgerufen werden können.

```
// Read JSON file content
String content = new String(Files.readAllBytes(Paths.get(jsonFileName)));
JSONArray jsonArray = new JSONArray(content);

// Convert JSON articles to Documents
List < Document > restaurants = new ArrayList < > ();
for (int i = 0; i < jsonArray.length(); i++) {
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    Document doc = Document.parse(jsonObject.toString());
    restaurants.add(doc);
}
//insert documents in collection
InsertManyResult result = collection.insertMany(restaurants);

System.out.println("Count of inserted documents: " + result.getInsertedIds().size());
```

`bulkWrite()`

Die `bulkWrite()` Methode ermöglicht es, mehrere Schreiboperationen (Einfügen, Aktualisieren, Löschen) in einem einzigen Stapel durchzuführen. Sie können `bulkWrite()` sie verwenden, wenn Sie verschiedene Arten von Vorgängen in einem einzigen Stapel ausführen müssen, z. B. wenn Sie einige Dokumente einfügen und andere aktualisieren müssen. `bulkWrite()`unterstützt zwei Arten von Batch-Schreibvorgängen, geordnet und ungeordnet:

- Geordnete Operationen — (Standard) Amazon DocumentDB verarbeitet die Schreibvorgänge sequentiell und stoppt beim ersten Fehler, auf den es stößt. Dies ist nützlich, wenn die Reihenfolge der Operationen wichtig ist, z. B. wenn spätere Operationen von früheren Vorgängen abhängen. Geordnete Operationen sind jedoch im Allgemeinen langsamer als ungeordnete Operationen. Bei geordneten Vorgängen müssen Sie den Fall berücksichtigen, dass der Stapel beim ersten Fehler stoppt und einige Operationen möglicherweise unbearbeitet bleiben.
- Ungeordnete Operationen — Ermöglicht Amazon DocumentDB, Einfügungen als einzelne Ausführung in der Datenbank zu verarbeiten. Wenn bei einem Dokument ein Fehler auftritt, wird der Vorgang mit den übrigen Dokumenten fortgesetzt. Dies ist besonders nützlich, wenn Sie große Datenmengen einfügen und einige Fehler tolerieren können, z. B. bei der Datenmigration oder bei Massenimporten, bei denen einige Dokumente aufgrund doppelter Schlüssel fehlschlagen können. Bei ungeordneten Vorgängen müssen Sie teilweise Erfolgsszenarien berücksichtigen, in denen einige Operationen erfolgreich sind, während andere fehlschlagen.

Bei der Arbeit mit der `bulkWrite()` Methode sind einige wichtige Klassen erforderlich. Erstens dient die `WriteModel`Klasse als Basisklasse für alle Schreiboperationen und für spezifische Implementierungen wie `InsertOneModel`,, `UpdateOneModelUpdateManyModelDeleteOneModel`, und `DeleteManyModel`behandelt verschiedene Arten von Operationen.

Die `BulkWriteOptions`Klasse ist notwendig, um das Verhalten von Massenoperationen zu konfigurieren, z. B. die ordered/unordered Ausführung festzulegen oder die Dokumentenvalidierung zu umgehen. Die `BulkWriteResult`Klasse bietet detaillierte Informationen zu den Ausführungsergebnissen, einschließlich der Anzahl der eingefügten, aktualisierten und gelöschten Dokumente.

Für die Fehlerbehandlung ist die `MongoBulkWriteException`Klasse von entscheidender Bedeutung, da sie Informationen über alle Fehler während des Massenvorgangs enthält, während die `BulkWriteError`Klasse spezifische Details zu einzelnen Vorgangsfehlern enthält. Der folgende Code zeigt ein Beispiel für das Einfügen einer Liste von Dokumenten sowie das Aktualisieren und Löschen eines einzelnen Dokuments, alles innerhalb der Ausführung eines einzigen `bulkWrite()` Methodenaufrufs. Der Code zeigt auch `BulkWriteResult`, wie mit `BulkWriteOptions`und gearbeitet wird und wie die `bulkWrite()` Operation korrekt mit Fehlern umgeht.

```
List < WriteModel < Document >> bulkOperations = new ArrayList < > ();
// get list of 10 documents representing 10 restaurants
List < Document > restaurantsToInsert = getSampleData();
```

```
for (Document doc: restaurantsToInsert) {
    bulkOperations.add(new InsertOneModel < > (doc));
}
// Update operation
bulkOperations.add(new UpdateOneModel < > (
    new Document("restaurantId", "REST-Y2E9H5"),
    new Document("", new Document("stats.likes", 20))
    .append("", new Document("rating.average", 4.5))));
// Delete operation
bulkOperations.add(new DeleteOneModel < > (new Document("restaurantId", "REST-
D2L431")));

// Perform bulkWrite operation
try {
    BulkWriteOptions options = new BulkWriteOptions()
        .ordered(false); // Allow unordered inserts

    BulkWriteResult result = collection.bulkWrite(bulkOperations, options);

    System.out.println("Inserted: " + result.getInsertedCount());
    System.out.println("Updated: " + result.getModifiedCount());
    System.out.println("Deleted: " + result.getDeletedCount());
} catch (MongoBulkWriteException e) {
    System.err.println("Bulk write error occurred: " + e.getMessage());
    // Log individual write errors
    for (BulkWriteError error: e.getWriteErrors()) {
        System.err.printf("Error at index %d: %s (Code: %d)%n", error.getIndex(),
 error.getMessage(),
            error.getCode());

        // Log the problematic document
        Document errorDoc = new Document(error.getDetails());
        if (errorDoc != null) {
            System.err.println("Problematic document: " + errorDoc);
        }
    }
} catch (Exception e) {
    System.err.println("Error during bulkWrite: " + e.getMessage());
}
```

Wiederholbare Schreibvorgänge

Im Gegensatz zu MongoDB unterstützt Amazon DocumentDB keine wiederholbaren Schreibvorgänge. Daher müssen Sie in ihren Anwendungen eine benutzerdefinierte Wiederholungslogik implementieren, insbesondere zur Behandlung von Netzwerkproblemen oder vorübergehender Nichtverfügbarkeit von Diensten. Eine gut implementierte Wiederholungsstrategie besteht in der Regel darin, die Verzögerung zwischen Wiederholungsversuchen zu erhöhen und die Gesamtzahl der Wiederholungsversuche zu begrenzen. Fehlerbehandlung mit WiederholungslogikIm Folgenden finden Sie ein Codebeispiel für die Erstellung einer Wiederholungslogik mit Fehlerbehandlung.

Lesen und Abrufen von Daten aus einer DocumentDB-Sammlung

Das Abfragen von Dokumenten in Amazon DocumentDB basiert auf mehreren Schlüsselkomponenten, mit denen Sie Daten präzise abrufen und bearbeiten können. Die `find()`Methode ist die grundlegende Abfrage APIs in MongoDB-Java-Treibern. Sie ermöglicht den komplexen Datenabruf mit zahlreichen Optionen zum Filtern, Sortieren und Projizieren von Ergebnissen. Neben der `find()` Methode `FindIterable`sind dies zwei weitere grundlegende Komponenten, `Filters`die die Bausteine für Abfrageoperationen in MongoDB-Java-Treibern bereitstellen.

Die `Filters` Klasse ist eine Hilfsklasse im MongoDB-Java-Treiber, die eine flüssige API zum Erstellen von Abfragefiltern bereitstellt. Diese Klasse bietet statische Factory-Methoden, die Instanzen von `Bson` Objekten erstellen, die verschiedene Abfragebedingungen repräsentieren. Zu den am häufigsten verwendeten Methoden gehören `eq()` für Gleichheitsvergleiche`gt()`,`lt()`,`gte()`, und `lte()` für numerische Vergleiche `and()` und `or()` für die Kombination mehrerer Bedingungen `in()` sowie `nin()` für Array-Zugehörigkeitstests und `regex()` für den Mustervergleich. Die Klasse ist typsicher konzipiert und bietet im Vergleich zu Abfragen, die auf Rohdokumenten basieren, eine bessere Überprüfung bei der Kompilierung, was sie zum bevorzugten Ansatz für die Erstellung von DocumentDB-Abfragen in Java-Anwendungen macht. Die Fehlerbehandlung ist robust und bei ungültigen Filterkonstruktionen werden eindeutige Ausnahmen ausgelöst.

`FindIterable`ist eine spezielle Schnittstelle, die für die Verarbeitung des Ergebnisses der `find()` Methode entwickelt wurde. Es bietet eine Vielzahl von Methoden zur Verfeinerung und Steuerung der Abfrageausführung und bietet eine flüssige API für die Methodenverkettung. Die Schnittstelle umfasst grundlegende Methoden zur Änderung von Abfragen, z. B. `limit()` zur Beschränkung der Anzahl der zurückgegebenen Dokumente, `skip()` zur Seitennummerierung, zur Reihenfolge der Ergebnisse, `sort()` zur Auswahl bestimmter Felder und `projection()` zur Indexauswahl. `hint()` Die Batch-, Skip- und Limit-Operationen in `FindIterable` sind wichtige Tools zur

Seitennummerierung und Datenverwaltung, mit deren Hilfe gesteuert werden kann, wie Dokumente aus der Datenbank abgerufen und verarbeitet werden.

Batching (`batchSize`) steuert, wie viele Dokumente DocumentDB in einem einzigen Netzwerk-Roundtrip an den Client zurückgibt. Wenn Sie eine Stapelgröße festlegen, gibt DocumentDB nicht alle passenden Dokumente auf einmal zurück, sondern gibt sie in Gruppen mit der angegebenen Stapelgröße zurück.

Mit Skip können Sie den Startpunkt Ihrer Ergebnisse verschieben und DocumentDB im Wesentlichen anweisen, eine bestimmte Anzahl von Dokumenten zu überspringen, bevor Treffer zurückgegeben werden. Beispielsweise `skip(20)` werden die ersten 20 übereinstimmenden Dokumente umgangen. Dies wird häufig in Paginierungsszenarien verwendet, in denen Sie nachfolgende Ergebnisseiten abrufen möchten.

Limit schränkt die Gesamtzahl der Dokumente ein, die von einer Abfrage zurückgegeben werden können. Wenn Sie angeben`limit(n)`, beendet DocumentDB die Rückgabe von Dokumenten, nachdem 'n' Dokumente zurückgegeben wurden, auch wenn es mehr Treffer in der Datenbank gibt.

`FindIterable`unterstützt sowohl Iterator- als auch Cursormuster beim Abrufen von Dokumenten aus Amazon DocumentDB. Der Vorteil der Verwendung `FindIterable` als Iterator besteht darin, dass Dokumente verzögert geladen werden können und Dokumente nur abgerufen werden, wenn sie von der Anwendung angefordert werden. Ein weiterer Vorteil der Verwendung von Iterator besteht darin, dass Sie nicht für die Aufrechterhaltung der Verbindung zum Cluster verantwortlich sind und daher kein explizites Schließen der Verbindung erforderlich ist.

`FindIterable`bietet auch Unterstützung für `MongoCursor`die Verwendung von Cursormustern bei der Arbeit mit Amazon DocumentDB DocumentDB-Abfragen. `MongoCursor`ist eine mongoDB-Java-Treiberspezifische Implementierung, die die Kontrolle über Datenbankoperationen und Ressourcenmanagement bietet. Sie implementiert die `AutoCloseable` Schnittstelle und ermöglicht so eine explizite Ressourcenverwaltung über try-with-resources Blöcke, was für das ordnungsgemäße Schließen von Datenbankverbindungen und die Freigabe von Serverressourcen von entscheidender Bedeutung ist. Standardmäßig läuft der Cursor innerhalb von 10 Minuten ab, und DocumentDB bietet Ihnen nicht die Möglichkeit, dieses Timeout-Verhalten zu ändern. Achten Sie bei der Arbeit mit Batchdaten darauf, den nächsten Datenstapel abzurufen, bevor der Cursor das Timeout erreicht. Ein wichtiger Aspekt bei der Verwendung `MongoCursor` ist, dass ein explizites Schließen erforderlich ist, um Ressourcenlecks zu verhindern.

In diesem Abschnitt werden mehrere Beispiele für`find()`, `Filters` und vorgestelltFindIterable.

Das folgende Codebeispiel zeigt, wie Sie `find()` mithilfe des Felds „RestaurantID“ ein einzelnes Dokument abrufen können:

```
Document filter = new Document("restaurantId", "REST-21G145");
Document result = collection.find(filter).first();
```

Die Verwendung `Filters` ermöglicht zwar eine bessere Fehlerprüfung bei der Kompilierung, mit dem Java-Treiber können Sie jedoch auch einen `Bson` Filter direkt in der `find()` Methode angeben. Der folgende Beispielcode übergibt `Bson` das Dokument an`find()`:

```
result = collection.find(new Document("$and", Arrays.asList(
    new Document("rating.totalReviews", new Document("$gt", 1000)),
    new Document("priceRange", "$$"))))
```

Der nächste Beispielcode zeigt mehrere Beispiele für die Verwendung der `Filters` Klasse mit`find()`:

```
FindIterable < Document > results;

// Exact match
results = collection.find(Filters.eq("name", "Thai Curry Palace"));

// Not equal
results = collection.find(Filters.ne("cuisine", "Thai"));

// find an element in an array
results = collection.find(Filters.in("features", Arrays.asList("Private Dining")));

// Greater than
results = collection.find(Filters.gt("rating.average", 3.5));

// Between (inclusive)
results = collection.find(Filters.and(
    Filters.gte("rating.totalReviews", 100),
    Filters.lte("rating.totalReviews", 200)));
// AND
results = collection.find(Filters.and(
    Filters.eq("cuisine", "Thai"),
    Filters.gt("rating.average", 4.5)));

// OR
results = collection.find(Filters.or(
```

```
    Filters.eq("cuisine", "Thai"),
    Filters.eq("cuisine", "American")));


// All document where the Field exists
results = collection.find(Filters.exists("michelin"));

// Regex
results = collection.find(Filters.regex("name", Pattern.compile("Curry",
 Pattern.CASE_INSENSITIVE)));

// Find all document where the array contain the list of value regardless of its order
results = collection.find(Filters.all("features", Arrays.asList("Private Dining",
 "Parking")));

// Array size
results = collection.find(Filters.size("features", 4));
```

Das folgende Beispiel zeigt, wie die Operationen von`sort()`, `skip()``limit()`, und `batchSize()` an einem `FindIterable` Objekt verkettet werden. Die Reihenfolge, in der diese Operationen ausgeführt werden, beeinflusst die Leistung Ihrer Abfrage. Als bewährte Methode sollte die Reihenfolge dieser Operationen`sort()`, `projection()``skip()`, `limit()` und lauten`batchSize()`.

```
FindIterable < Document > results = collection.find(Filters.gt("rating.totalReviews",
 1000))
    // Sorting
    .sort(Sorts.orderBy(
        Sorts.descending("address.city"),
        Sorts.ascending("cuisine")))
    // Field selection
    .projection(Projections.fields(
        Projections.include("name", "cuisine", "priceRange"),
        Projections.excludeId()))

    // Pagination
    .skip(20)
    .limit(10)
    .batchSize(2);
```

Der folgende Beispielcode zeigt die Erstellung eines Iterators am`FindIterable`. Es verwendet das `forEach` Java-Konstrukt, um die Ergebnismenge zu durchlaufen.

```
collection.find(Filters.eq("cuisine", "American")).forEach(doc ->
 System.out.println(doc.toJson()));
```

Im letzten `find()` Codebeispiel wird gezeigt, wie es zum Abrufen von Dokumenten verwendet `cursor()` wird. Es erstellt den Cursor im Try-Block, wodurch sichergestellt wird, dass der Cursor geschlossen wird, wenn der Code den Try-Block verlässt.

```
try (MongoCursor < Document > cursor = collection.find(Filters.eq("cuisine",
 "American"))
    .batchSize(25)
    .cursor()) {
    while (cursor.hasNext()) {
        Document doc = cursor.next();
        System.out.println(doc.toJson());
    }
} // Cursor automatically closed
```

Aktualisieren vorhandener Dokumente in einer DocumentDB-Sammlung

Amazon DocumentDB bietet flexible und leistungsstarke Mechanismen zum Ändern vorhandener Dokumente und zum Einfügen neuer Dokumente, wenn sie nicht existieren. Der MongoDB-Java-Treiber bietet mehrere Methoden für Updates: `updateOne()` für die Aktualisierung einzelner Dokumente, `updateMany()` für die Aktualisierung mehrerer Dokumente und `replaceOne()` für den vollständigen Austausch von Dokumenten. Neben diesen drei Methoden `UpdateResult`sind, `UpdatesUpdateOptions`, und weitere grundlegende Komponenten, die die Bausteine für Aktualisierungsvorgänge in MongoDB-Java-Treibern bereitstellen.

Die `Updates` Klasse im MongoDB-Java-Treiber ist eine Hilfsklasse, die statische Factory-Methoden zum Erstellen von Aktualisierungsoperatoren bereitstellt. Sie dient als primärer Builder für die typsichere und lesbare Konstruktion von Aktualisierungsoperationen. Grundlegende Methoden wie `set()unset()`, und `inc()` ermöglichen die direkte Änderung der Dokumente. Die Leistungsfähigkeit dieser Klasse wird deutlich, wenn mehrere Operationen mit der `Updates.combine()` Methode kombiniert werden, mit der mehrere Aktualisierungsvorgänge atomar ausgeführt werden können, wodurch die Datenkonsistenz gewährleistet wird.

`UpdateOptions`ist eine leistungsstarke Konfigurationsklasse im Java-Treiber von MongoDB, die grundlegende Anpassungsmöglichkeiten für Dokumentaktualisierungsvorgänge bietet. Zwei wichtige Aspekte dieser Klasse sind die Unterstützung von Upsert- und Array-Filtern für Aktualisierungsvorgänge. Die Upsert-Funktion, aktiviert durch`upsert(true)`, ermöglicht die

Erstellung neuer Dokumente, wenn während eines Aktualisierungsvorgangs keine passenden Dokumente gefunden werden. Durch `arrayFilters()` den Aktualisierungsvorgang können Array-Elemente, die bestimmte Kriterien erfüllen, präzise aktualisiert werden.

`UpdateResult`Der Java-Treiber von MongoDB bietet den Feedback-Mechanismus, der das Ergebnis eines Aktualisierungsvorgangs detailliert beschreibt. Diese Klasse umfasst drei wichtige Kennzahlen: die Anzahl der Dokumente, die den Aktualisierungskriterien entsprechen (`matchedCount`), die Anzahl der tatsächlich geänderten Dokumente (`modifiedCount`) und Informationen über alle geänderten Dokumente (). `upsertedId` Das Verständnis dieser Metriken ist für die korrekte Fehlerbehandlung, die Überprüfung von Aktualisierungsvorgängen und die Aufrechterhaltung der Datenkonsistenz in Anwendungen unerlässlich.

Aktualisieren und ersetzen Sie ein einzelnes Dokument

In DocumentDB kann die Aktualisierung eines einzelnen Dokuments mit der Methode updateOne () durchgeführt werden. Diese Methode verwendet einen Filterparameter, der normalerweise von der `Filters` Klasse bereitgestellt wird, um das zu aktualisierende Dokument zu identifizieren, einen `Updat` e-Parameter, der bestimmt, welche Felder aktualisiert werden sollen, und einen optionalen `UpdateOptions` Parameter, um verschiedene Optionen für die Aktualisierung festzulegen. Bei Verwendung `updateOne()` dieser Methode wird nur das erste Dokument aktualisiert, das den Auswahlkriterien entspricht. Der folgende Beispielcode aktualisiert ein einzelnes Feld eines Dokuments:

```
collection.updateOne(Filters.eq("restaurantId", "REST-Y2E9H5"),
    Updates.set("name", "Amazing Japanese sushi"));
```

Um mehrere Felder in einem Dokument zu aktualisieren, verwenden Sie `updateOne()` with `Update.combine()` wie im folgenden Beispiel gezeigt. Dieses Beispiel zeigt auch, wie ein Element zu einem Array im Dokument hinzugefügt wird.

```
List<Bson> updates = new ArrayList<>();
// Basic field updates
updates.add(Updates.set("name", "Shanghai Best"));
// Array operations
updates.add(Updates.addEachToSet("features", Arrays.asList("Live Music")));
// Counter updates
updates.add(Updates.inc("rating.totalReviews", 10));
// Combine all updates
Bson combinedUpdates = Updates.combine(updates);
```

```
// Execute automic update with one call
collection.updateOne(Filters.eq("restaurantId","REST-1J83NH"), combinedUpdates);
```

Das folgende Codebeispiel zeigt, wie ein Dokument in der Datenbank aktualisiert wird. Wenn das angegebene Dokument nicht existiert, wird es stattdessen automatisch als neues Dokument eingefügt. Dieser Code zeigt auch, wie die über das `UpdateResult` Objekt verfügbaren Metriken verwendet werden.

```
Bson filter = Filters.eq("restaurantId", "REST-0Y9GL0");
Bson update = Updates.set("cuisine", "Indian");
// Upsert operation
UpdateOptions options = new UpdateOptions().upsert(true);
UpdateResult result = collection.updateOne(filter, update, options);

if (result.getUpsertedId() != null) {
    System.out.println("Inserted document with _id: " + result.getUpsertedId());
} else {
     System.out.println("Updated " + result.getModifiedCount() + " document(s)");
}
```

Das folgende Codebeispiel zeigt, wie Sie mithilfe der `replaceOne()` Methode ein vorhandenes Dokument vollständig durch ein neues Dokument ersetzen können, anstatt einzelne Felder zu aktualisieren. Die `replaceOne()` Methode überschreibt das gesamte Dokument, wobei nur das `_id` Feld des Originals beibehalten wird. Wenn mehrere Dokumente den Filterkriterien entsprechen, wird nur das zuerst gefundene Dokument ersetzt.

```
Document newDocument = new Document()
                .append("restaurantId", "REST-0Y9GL0")
                .append("name", "Bhiryani Adda")
                .append("cuisine", "Indian")
                .append("rating", new Document()
                        .append("average", 4.8)
                        .append("totalReviews", 267))
                .append("features", Arrays.asList("Outdoor Seating", "Live Music"));

UpdateResult result = collection.replaceOne(
                    Filters.eq("restaurantId", "REST-0Y9GL0"),
                    newDocument);
System.out.printf("Modified %d document%n", result.getModifiedCount());
```

Aktualisieren Sie mehrere Dokumente

Es gibt zwei Möglichkeiten, mehrere Dokumente in einer Sammlung gleichzeitig zu aktualisieren. Sie können die `updateMany()` Methode oder die `UpdateManyModel`mit der `bulkWrite()` Methode verwenden. Die `updateMany()` Methode verwendet einen Filterparameter, um Dokumente für die Aktualisierung auszuwählen, den `Update` Parameter zur Identifizierung der zu aktualisierenden Felder und einen optionalen `UpdateOptions` Parameter zur Angabe von Aktualisierungsoptionen.

Der folgende Beispielcode demonstriert die Verwendung der `updateMany()` Methode:

```
Bson filter = Filters.and(
    Filters.in("features", Arrays.asList("Private Dining")),
    Filters.eq("cuisine", "Thai"));
UpdateResult result1 = collection.updateMany(filter, Updates.set("priceRange", "$$$"));
```

Der folgende Beispielcode demonstriert die `bulkWrite()` Methode mit demselben Update:

```
BulkWriteOptions options = new BulkWriteOptions().ordered(false);
List < WriteModel < Document >> updates = new ArrayList < > ();
Bson filter = Filters.and(
    Filters.in("features", Arrays.asList("Private Dining")),
    Filters.eq("cuisine", "Indian"));
Bson updateField = Updates.set("priceRange", "$$$");
updates.add(new UpdateManyModel < > (filter, updateField));
BulkWriteResult result = collection.bulkWrite(updates, options);
System.out.printf("Modified %d document%n", result.getModifiedCount());
```

Dokumente aus einer DocumentDB-Sammlung entfernen

Der MongoDB-Java-Treiber bietet `deleteOne()` die Möglichkeit, ein einzelnes Dokument und mehrere Dokumente `deleteMany()` zu entfernen, die bestimmten Kriterien entsprechen. Genau wie beim Aktualisieren kann auch der Löschvorgang mit der `bulkWrite()` Methode verwendet werden. Beides `deleteOne()` und gibt ein `DeleteResult`Objekt `deleteMany()` zurück, das Informationen über das Ergebnis des Vorgangs enthält, einschließlich der Anzahl der gelöschten Dokumente. Im Folgenden finden Sie ein Beispiel für die Verwendung `deleteMany()` zum Entfernen mehrerer Dokumente:

```
Bson filter = Filters.and(
    Filters.eq("cuisine", "Thai"),
    Filters.lt("rating.totalReviews", 50));
```

```
DeleteResult result = collection.deleteMany(filter);
System.out.printf("Deleted %d document%n", result.getDeletedCount());
```

Fehlerbehandlung mit Wiederholungslogik

Eine robuste Strategie zur Fehlerbehandlung für Amazon DocumentDB sollte die Kategorisierung von Fehlern in wiederholbare Fehler (wie Netzwerk-Timeouts, Verbindungsprobleme) und nicht wiederholbare Fehler (wie Authentifizierungsfehler, ungültige Abfragen) implementieren. Bei Betriebsfehlern aufgrund von Fehlern, die wiederholt werden sollten, sollte eine Zeitverzögerung zwischen den einzelnen Wiederholungsversuchen sowie die maximale Anzahl von Wiederholungsversuchen vorgesehen werden. Die CRUD-Operationen sollten sich in einem Try-Catch-Block befinden, der Catches `MongoException`und seine Unterklassen enthält. Darüber hinaus sollte es die Überwachung und Protokollierung von Fehlern beinhalten, um die betriebliche Transparenz zu gewährleisten. Im Folgenden finden Sie einen Beispielcode, der zeigt, wie die Behandlung von Wiederholungsfehlern implementiert wird:

```
int MAX_RETRIES = 3;
int INITIAL_DELAY_MS = 1000;
int retryCount = 0;

while (true) {
    try {
        crud_operation(); //perform crud that will throw MongoException or one of its
 subclass
        break;
    } catch (MongoException e) {
        if (retryCount < MAX_RETRIES) {
            retryCount++;
            long delayMs = INITIAL_DELAY_MS * (long) Math.pow(2, retryCount - 1);
            try {
                TimeUnit.MILLISECONDS.sleep(delayMs);
            } catch (InterruptedException t) {
                Thread.currentThread().interrupt();
                throw new RuntimeException("Retry interrupted", t);
            }
            continue;
        } else
            throw new RuntimeException("Crud operation failed", e);
    }
}
```

Indexverwaltung in Amazon DocumentDB mit Java

Indizes ermöglichen das effiziente Abrufen von Daten aus einer Amazon DocumentDB-Sammlung. Ohne Indizes muss DocumentDB jedes Dokument in der Sammlung scannen, um Ergebnisse zurückzugeben, die einer bestimmten Abfrage entsprechen. Dieses Thema enthält Informationen zum Erstellen, Löschen und Auflisten von Indizes mithilfe der MongoDB-Java-Treiber. Außerdem wird erläutert, wie ermittelt werden kann, ob ein bestimmter Index in der Abfrage verwendet wird, und wie Amazon DocumentDB Hinweise zur Verwendung eines bestimmten Indexes gegeben werden kann.

Themen

- Indizes mit Java erstellen
- Indizes löschen
- Festlegung der Indexauswahl und Bereitstellung eines Indexhinweises

Amazon DocumentDB unterstützt viele Arten von Indizes. Einen umfassenden Überblick über alle unterstützten Indizes finden Sie in diesem Blogbeitrag.

Indizes mit Java erstellen

Es gibt zwei Mechanismen zum Erstellen von Indizes in Amazon DocumentDB mithilfe von MongoDB-Java-Treibern: durch `runCommand()` und entweder durch die `createIndex()` Methode für einen einzelnen Index oder durch die `createIndexes()` Methode für mehrere Indizes. Ein Grund für die Verwendung der `createIndexes()` Methoden `createIndex()` und besteht darin, dass Sie eine bessere Fehlerbehandlung aufbauen können, indem Sie bestimmte Fehler im Zusammenhang mit der Indexerstellung abfangen. Ein weiterer Grund für die Verwendung dieser Methode `runCommand()` ist, dass der MongDB-Java-Treiber eine Vielzahl von unterstützenden Klassen für die Indexerstellung und -manipulation bereitstellt. Beachten Sie, dass diese unterstützenden Klassen nur verwendet werden können, wenn Sie die Methoden `createIndex()` oder `createIndexes()` verwenden. Es gibt drei unterstützende Klassen:

- **`Indexes`**— Diese Klasse dient als Hilfsklasse und bietet statische Factory-Methoden zur Erstellung verschiedener Arten von Indizes. Sie vereinfacht den Prozess der Erstellung komplexer Indexdefinitionen und wird häufig in Verbindung mit anderen indexbezogenen Klassen verwendet.
- **`IndexModel`**— Dies ist eine grundlegende Klasse, die sowohl die Definition der Indexschlüssel als auch ihre Optionen kapselt. Sie stellt eine vollständige Indexspezifikation dar, die kombiniert, was indexiert werden soll (die Schlüssel) und wie indexiert werden soll (die Optionen). Diese Klasse ist besonders nützlich, wenn Sie mehrere Indizes gleichzeitig erstellen, da Sie damit eine Sammlung

von Indexspezifikationen definieren können, die an die `createIndexes()` Methode übergeben werden können.

- **`IndexOptions`**— Dies ist eine umfassende Konfigurationsklasse, die eine Vielzahl von Methoden zur Anpassung des Indexverhaltens bietet. Sie umfasst Einstellungen für eindeutige Indizes, Sparse-Indizes, Ablaufzeit (TTL) und partielle Filterausdrücke. Durch Methodenverkettung können Sie mehrere Optionen wie die Indexerstellung im Hintergrund und eindeutige Einschränkungen konfigurieren.

Erstellen Sie einen einzelnen Index

Dieses Beispiel zeigt, wie ein einzelner Index mit der Methode `createIndex( ()` im Hintergrund erstellt wird. Informationen zur Indexerstellung im Hintergrund und Vordergrund finden Sie unterTypen der Indexerstellung. Im folgenden Codebeispiel wird `IndexOptions`ein eindeutiger Index mit dem Namen „unique_restaurantID_IDX“ im Hintergrund erstellt. Dieses `IndexOptions` Objekt wird dann an die Methode übergeben. `createIndex()`

```
collection.createIndex(
    Indexes.ascending("restaurantId"),
    new IndexOptions()
        .unique(true)
        .name("unique_restaurantId_idx")
        .background(true));
```

Erstellen Sie mehrere Indizes

In diesem Beispiel werden mithilfe der Methode mehrere Indizes erstellt. `createIndexes()` Es erstellt zuerst die Option für jeden Index mithilfe des `IndexModel`Objekts und übergibt dann eine Liste von `IndexModel` Objekten an die `createIndexes()` Methode. Das folgende Codebeispiel zeigt, wie ein zusammengesetzter Index mithilfe der `Indexes`Utility-Klasse erstellt wird. Diese Klasse wird auch verwendet, um anzugeben, ob Sie einen Index mit aufsteigender oder absteigender Sortierreihenfolge erstellen möchten. Nach dem Erstellen mehrerer Indizes überprüft sie die Indexerstellung, indem sie die Methode aufruft. `listIndexes()`

```
// Single Field Index on cuisine
IndexModel singleIndex = new IndexModel(
    Indexes.ascending("cuisine"),
    new IndexOptions().name("cuisine_idx"));
```

```
// Compound Index
IndexModel compoundIndex = new IndexModel(
    Indexes.compoundIndex(
        Indexes.ascending("address.state"),
        Indexes.ascending("priceRange")),
    new IndexOptions().name("location_price_idx"));

// Build a list of IndexModel for the indexes
List < IndexModel > indexes = Arrays.asList(
    singleIndex,
    compoundIndex
);

collection.createIndexes(indexes);

// Verify created indexes
collection.listIndexes().forEach(index - > System.out.println("Created index: " +
 index.toJson()));
```

Erstellen Sie dünnbesetzte Indizes und partielle Indizes

Dieses Beispiel zeigt, dass ein dünnbesetzter Index und ein partieller Index erstellt werden, indem `IndexModel`für jeden Indextyp ein Index erstellt wird.

```
// Sparse Index Model, this will identify only those documents that have a
// michelin star rating
IndexModel sparseIndex = new IndexModel(
    Indexes.ascending("michelin.star"),
    new IndexOptions()
    .name("michelin_sparse_idx")
    .sparse(true));

// Partial Index Model where the restaurant is active and has a rating of 4 and above
IndexModel partialIndex = new IndexModel(
    Indexes.ascending("rating.average"),
    new IndexOptions()
    .name("high_rated_active_idx")
    .partialFilterExpression(
        Filters.and(
            Filters.eq("isActive", true),
            Filters.gte("rating.average", 4.0))));
```

Erstellen Sie einen Textindex

Dieses Beispiel zeigt, wie ein Textindex erstellt wird. In einer Sammlung ist nur ein Textindex zulässig, aber dieser eine Textindex kann ein zusammengesetzter Index sein, der mehrere Felder abdeckt. Wenn Sie mehrere Felder im Textindex verwenden, können Sie auch jedem Feld im Index Gewichtungen zuweisen. Textindizes für Array-Felder werden von Amazon DocumentDB nicht unterstützt, und obwohl Sie bis zu 30 Felder im zusammengesetzten Textindex verwenden können, kann nur drei Feldern eine Gewichtung zugewiesen werden.

```
IndexModel textIndex = new IndexModel(
    new Document()
    .append("name", "text")
    .append("description", "text")
    .append("cuisine", "text"),
    new IndexOptions()
    .name("restaurant_text_idx")
    .weights(new Document()
        .append("name", 10) // Restaurant name gets highest weight
        .append("description", 5) // Description get medium weight
        .append("cuisine", 2) // Cuisine type gets low weight
    ));

collection.createIndex(textIndex.getKeys(), textIndex.getOptions());
```

Erstellen Sie einen Index mit **`runCommand()`**

Amazon DocumentDB unterstützt die parallel Indexerstellung, um den Zeitaufwand für die Erstellung von Indizes zu verringern. Bei der parallelen Indizierung werden mehrere Worker gleichzeitig verwendet. Für die Indexerstellung werden standardmäßig zwei Worker verwendet. Dieser Blogbeitrag bietet eine eingehende Diskussion zur parallel Indizierung. Derzeit unterstützen MongDB-Java-Treiber die Angabe der Worker-Option nicht, wenn Sie `createIndex()` oder `createIndexes()` verwenden. Daher können Sie Worker nur über die angeben. `runCommand` Das folgende Codebeispiel zeigt, wie Sie `runCommand` einen Index erstellen, der den Worker auf vier erhöht:

```
Document command = new Document("createIndexes", "Restaurants")
    .append("indexes", Arrays.asList(
        new Document("key", new Document("name", 1))
        .append("name", "restaurant_name_idx")
        .append("workers", 4) // Specify number of workers
    ));
```

```
Document commendResult = connectedDB.runCommand(command);
```

Indizes löschen

Der MongoDB-Java-Treiber bietet mehrere Methoden zum Löschen von Indizes, die auf unterschiedliche Szenarien und Ihre Präferenzen zugeschnitten sind. Sie können Indizes nach Namen, nach Schlüsselspezifikation oder alle Indizes auf einmal löschen. Die Methoden `dropIndex()` und `dropIndexes()` können für ein Auflistungsobjekt aufgerufen werden, um einen Index zu löschen. Wenn Sie einen Index nach Namen löschen, sollten Sie sicherstellen, dass der richtige Indexname verwendet wird, was möglicherweise nicht immer intuitiv ist, insbesondere bei zusammengesetzten oder automatisch generierten Indizes. Der Versuch, einen nicht existierenden Index zu löschen, führt zu einem. `MongoCommandException` Der `default _id` Index kann nicht gelöscht werden, da er die Einzigartigkeit des Dokuments innerhalb der Sammlung sicherstellt.

Das folgende Codebeispiel zeigt, wie ein Index gelöscht wird, indem der Feldname angegeben wird, in dem der Index erstellt wurde, oder indem alle Indizes gelöscht werden:

```
String indexName = "unique_restaurantId_idx";
Document keys = new Document("cuisine", 1);
// Drop index by name
collection.dropIndex(indexName);

// Drop index by keys
collection.dropIndex(keys);

// Drop all indexes
collection.dropIndexes();
```

Stellen Sie beim Löschen von Indizes mit mehreren Schlüsseln sicher, dass ein zusammengesetzter Index vorhanden ist, der alle angegebenen Schlüssel enthält und dass die Reihenfolge der Schlüssel korrekt ist. Der obige Beispielcode zur Indexerstellung zeigt einen zusammengesetzten Schlüssel für „Küche“ und Funktionen. Wenn Sie versuchen, diesen zusammengesetzten Schlüssel zu löschen, aber die Reihenfolge nicht der Reihenfolge entspricht, in der er erstellt wurde, tritt ein MongoCommnadException Fehler wie folgt auf:

```
Document keys = new Document("features", 1)
    .append("cuisine", 1);
try {
    // Drop index by keys
    collection.dropIndex(keys);
```

```
    System.out.println("Successfully dropped index with keys: " + keys.toJson());

} catch (MongoCommandException commErr) {
    System.out.println("Error dropping index: " + commErr.getErrorMessage());
    throw new RuntimeException("MongoCommandException was thrown while dropping index",
 commErr);
}
```

Der folgende Fehler wird angezeigt:

```
Error dropping index: Cannot drop index: index not found.
Tests run: 3, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.819 sec <<< FAILURE!
com.amazon.docdb.guide.DocDBGuideTest.testindexGuide()  Time elapsed: 0.817 sec  <<<
 FAILURE!
org.opentest4j.AssertionFailedError: Unexpected exception thrown:
 java.lang.RuntimeException: MongoCommandException was thrown while dropping index
```

Festlegung der Indexauswahl und Bereitstellung eines Indexhinweises

Die Arbeit mit der Erklärungsfunktion in Amazon DocumentDB ist entscheidend, um die Abfrageleistung und die Indexnutzung zu verstehen. Wenn Sie eine Abfrage ausführen, können Sie die `explain()` Methode anhängen, um detaillierte Informationen über den Abfrageplan zu erhalten, einschließlich der verwendeten Indizes. Die `explain()` Ausgabe bietet Einblicke in die Phasen der Abfrageausführung, die Anzahl der untersuchten Dokumente und die für jede Phase benötigte Zeit. Diese Informationen sind von unschätzbarem Wert, um festzustellen, ob ein bestimmter Index effektiv verwendet wird oder ob die Abfrage von einer anderen Indexstruktur profitieren könnte.

Die `explain()` Methode kann mit der Methode verkettet werden. `find()` Die `explain()` Methode kann eine optionale `ExplainVerbosity`Aufzählung verwenden, die den Grad der Ausführlichkeit bestimmt, der von zurückgegeben wird. `explain()` Derzeit werden nur die `QUERY_PLANNER` Enumeratoren `EXECUTION_STATS` und von DocumentDB unterstützt. Das folgende Codebeispiel zeigt, wie der Abfrageplaner für eine bestimmte Abfrage abgerufen wird:

```
// Query we want to analyze
Document query = new Document()
    .append("cuisine", "Thai")
    .append("rating.average", new Document("$gte", 4.0));


Document allPlansExplain =
 collection.find(query).explain(ExplainVerbosity.QUERY_PLANNER);
```

```
System.out.println("All Plans Explain:\n" + allPlansExplain.toJson());
```

Das folgende JSON-Dokument wird für die Ausführlichkeitsstufe des Abfrageplaners zurückgegeben:

```
{
  "queryPlanner": {
    "plannerVersion": 1,
    "namespace": "ProgGuideData.Restaurants",
    "winningPlan": {
      "stage": "IXSCAN",
      "indexName": "cuisine_idx",
      "direction": "forward"
    }
  },
  "serverInfo": {
    "host": "guidecluster3",
    "port": 27017,
    "version": "5.0.0"
  },
  "ok": 1,
  "operationTime": {
    "$timestamp": {
      "t": 1739221668,
      "i": 1
    }
  }
}
```

Sie haben mehrere Möglichkeiten, Amazon DocumentDB zu beeinflussen oder zu zwingen, einen bestimmten Index zu verwenden. Mit den `hintString()` Methoden `hint()` und können Sie das Standardverhalten des Abfrageoptimierers bei der Indexauswahl außer Kraft setzen, indem Sie explizit angeben, welcher Index für eine Abfrage verwendet werden soll. Der Abfrageoptimierer von DocumentDB ist zwar generell eine gute Wahl für die Indexauswahl, es gibt jedoch auch Szenarien, in denen das `hint()` Durchsetzen eines bestimmten Indexes von Vorteil sein `hintString()` kann, z. B. beim Umgang mit verzerrten Daten oder beim Testen der Indexleistung.

Das folgende Codebeispiel erzwingt die Verwendung des zusammengesetzten Index „cuisine_features_idx“ für dieselbe Abfrage, die im obigen Code ausgeführt wurde:

```
// Query we want to analyze
Document query = new Document()
```

```
    .append("cuisine", "Thai")
    .append("rating.average", new Document("$gte", 4.0));

List < Document > queryDocs = new ArrayList < > ();
collection.find(query).hintString("cuisine_features_idx").forEach(doc - >
 queryDocs.add(doc));
```

Ereignisgesteuerte Programmierung mit Amazon DocumentDB und Java

Die ereignisgesteuerte Programmierung im Kontext von Amazon DocumentDB stellt ein leistungsstarkes Architekturmuster dar, bei dem Datenbankänderungen als primäre Ereignisgeneratoren dienen, die nachfolgende Geschäftslogik und -prozesse auslösen. Wenn Datensätze in eine DocumentDB-Auflistung eingefügt, aktualisiert oder gelöscht werden, wirken sich diese Änderungen als Ereignisse aus, die automatisch verschiedene nachgelagerte Prozesse, Benachrichtigungen oder Datensynchronisierungsaufgaben auslösen. Dieses Muster ist besonders nützlich in modernen verteilten Systemen, in denen mehrere Anwendungen oder Dienste in Echtzeit auf Datenänderungen reagieren müssen. Der Hauptmechanismus für die Implementierung ereignisgesteuerter Programmierung in DocumentDB sind Change-Streams.

> **Note**
>
> In diesem Handbuch wird davon ausgegangen, dass Sie Change-Streams für eine Sammlung aktiviert haben, mit der Sie arbeiten. Weitere Informationen Change-Streams mit Amazon DocumentDB verwenden zum Aktivieren von Change-Streams in der Sammlung finden Sie unter.

Arbeiten mit Change-Streams aus der Java-Anwendung

Die `watch()` Methode im Java-Treiber von MongoDB ist der wichtigste Mechanismus zur Überwachung von Datenänderungen in Echtzeit in Amazon DocumentDB. Die `watch()` Methode kann von Objekten `MongoClient` `MongoDatabase`, und `MongoCollection`aufgerufen werden.

Die `watch()` Methode gibt eine Instanz zurück `ChangeStreamIterable`, die verschiedene Konfigurationsoptionen unterstützt, darunter die vollständige Suche nach Aktualisierungen in Dokumenten, die Bereitstellung von Wiederaufnahmetokens und Zeitstempeln für die Zuverlässigkeit sowie Pipeline-Aggregationsphasen zum Filtern von Änderungen.

`ChangeStreamIterable`implementiert die Java-Kernschnittstelle `Iterable` und kann mit verwendet werden. `forEach()` Um Ereignisse mithilfe von zu erfassen`forEach()`, übergeben

Sie eine Callback-Funktion an`forEach()`, die das geänderte Ereignis verarbeitet. Der folgende Codeausschnitt zeigt, wie Sie einen Change-Stream für eine Sammlung öffnen, um die Überwachung von Änderungsereignissen zu starten:

```
ChangeStreamIterable < Document > iterator = collection.watch();
iterator.forEach(event - > {
    System.out.println("Received a change: " + event);
});
```

Eine andere Möglichkeit, alle Änderungsereignisse zu durchlaufen, besteht darin, einen Cursor zu öffnen, der eine Verbindung zum Cluster aufrechterhält und kontinuierlich neue Änderungsereignisse empfängt, sobald sie auftreten. Um einen Change-Streams-Cursor zu erhalten, verwenden Sie die `cursor()` Methode object. `ChangeStreamIterable` Das folgende Codebeispiel zeigt, wie Änderungsereignisse mithilfe des Cursors überwacht werden:

```
try (MongoChangeStreamCursor < ChangeStreamDocument < Document >> cursor =
 collection.watch().cursor()) {
    System.out.println(cursor.tryNext());
}
```

Es hat sich bewährt, entweder den `MongoChangeStreamCursor`in einer try-with-resource Anweisung zu erstellen oder den Cursor manuell zu schließen. Beim Aufrufen der `cursor()` Methode on wird eine `ChangeStreamIterable`zurückgegeben`MongoChangeStreamCursor`, die über einem `ChangeStreamDocument`Objekt erstellt wurde.

Die `ChangeStreamDocument`Klasse ist eine wichtige Komponente, die einzelne Änderungsereignisse im Stream darstellt. Sie enthält detaillierte Informationen zu jeder Änderung, einschließlich des Vorgangstyps (Einfügen, Aktualisieren, Löschen, Ersetzen), des Dokumentschlüssels, der Namespace-Informationen und des vollständigen Dokumentinhalts, sofern verfügbar. Die Klasse bietet Methoden für den Zugriff auf verschiedene Aspekte des Änderungsereignisses, z. B. `getOperationType()` für die Bestimmung der Art der Änderung, für den `getFullDocument()` Zugriff auf den vollständigen Status des Dokuments und `getDocumentKey()` für die Identifizierung des geänderten Dokuments.

Das `ChangeStreamDocument`Objekt stellt zwei wichtige Informationen bereit: ein Wiederaufnahme-Token und die Uhrzeit des Änderungsereignisses.

Resume-Tokens und zeitbasierte Operationen in DocumentDB-Change-Streams bieten wichtige Mechanismen für die Aufrechterhaltung der Kontinuität und die Verwaltung des Zugriffs auf

historische Daten. Ein Resume-Token ist eine eindeutige Kennung, die für jedes Änderungsereignis generiert wird. Sie dient als Lesezeichen, mit dem Anwendungen die Verarbeitung von Change-Streams nach Verbindungsabbrüchen oder Ausfällen an einem bestimmten Punkt wieder aufnehmen können. Wenn ein Change-Stream-Cursor erstellt wird, kann er über `resumeAfter()` diese Option ein zuvor gespeichertes Resume-Token verwenden, sodass der Stream dort weitergeführt werden kann, wo er aufgehört hat, anstatt am Anfang zu beginnen oder Ereignisse zu verlieren.

Zeitbasierte Operationen in Change-Streams bieten unterschiedliche Ansätze, um den Ausgangspunkt der Überwachung von Änderungsereignissen zu verwalten. Mit dieser `startAtOperationTime()` Option können Sie beginnen, Änderungen zu beobachten, die zu oder nach einem bestimmten Zeitstempel vorgenommen wurden. Diese zeitbasierten Funktionen sind besonders nützlich in Szenarien, in denen historische Daten verarbeitet, point-in-time wiederhergestellt oder zwischen Systemen synchronisiert werden müssen.

Im folgenden Codebeispiel wird das Ereignis abgerufen, das mit dem Einfügedokument verknüpft ist, dessen Resume-Token erfasst und dann dieses Token bereitgestellt, um mit der Überwachung auf Ereignisse nach dem Einfügeereignis zu beginnen. Das Ereignis wird dem Aktualisierungsereignis zugeordnet, ruft dann die Clusterzeit ab, zu der die Aktualisierung stattgefunden hat, und verwendet diesen Zeitstempel als Ausgangspunkt für die weitere Verarbeitung.

```
BsonDocument resumeToken;
BsonTimestamp resumeTime;

try (MongoChangeStreamCursor < ChangeStreamDocument < Document >> cursor =
 collection.watch().cursor()) {
    System.out.println("****************** Insert Document ******************");
    ChangeStreamDocument < Document > insertChange = cursor.tryNext();
    resumeToken = insertChange.getResumeToken();
    printJson(cursor.tryNext());
}
try (MongoChangeStreamCursor < ChangeStreamDocument < Document >> cursor =
 collection.watch()
    .resumeAfter(resumeToken)
    .cursor()) {
    System.out.println("****************** Update Document ******************");
    ChangeStreamDocument < Document > insertChange = cursor.tryNext();
    resumeTime = insertChange.getClusterTime();
    printJson(cursor.tryNext());
}
try (MongoChangeStreamCursor < ChangeStreamDocument < Document >> cursor =
 collection.watch()
```

```
    .startAtOperationTime(resumeTime)
    .cursor()) {
    System.out.println("****************** Delete Document ******************");
    printJson(cursor.tryNext());
  }
```

Standardmäßig umfasst das Aktualisierungsänderungsereignis nicht das gesamte Dokument und nur die vorgenommenen Änderungen. Wenn Sie auf das gesamte Dokument zugreifen müssen, das aktualisiert wurde, können Sie die `fullDocument()` Methode für das `ChangeStreamIterable`Objekt aufrufen. Denken Sie daran, dass, wenn Sie bei einem Aktualisierungsereignis die Rückgabe eines vollständigen Dokuments anfordern, das Dokument zurückgegeben wird, das zum Zeitpunkt des Aufrufs der Change-Streams vorhanden war.

Diese Methode verwendet eine `FullDocument`Aufzählung als Parameter. Derzeit unterstützt Amazon DocumentDB nur DEFAULT und UPDATE_LOOKUP Werte. Der folgende Codeausschnitt zeigt, wie Sie das vollständige Dokument für Aktualisierungsereignisse anfordern, wenn Sie beginnen, nach Änderungen zu suchen:

```
try (MongoChangeStreamCursor < ChangeStreamDocument < Document >> cursor =
 collection.watch().fullDocument(FullDocument.UPDATE_LOOKUP).cursor())
```

Verwenden der JSON-Schemavalidierung

Mithilfe des `$jsonSchema` Testabfrageoperators können Sie überprüfen, ob Dokumente in Ihre Sammlungen eingefügt werden.

Themen

- JSON-Schemavalidierung erstellen und verwenden
- Unterstützte Schlüsselwörter
- bypassDocumentValidation
- Einschränkungen

JSON-Schemavalidierung erstellen und verwenden

Eine Sammlung mit Schemavalidierung erstellen

Sie können eine Sammlung mit `createCollection` Betriebs- und Validierungsregeln erstellen. Diese Validierungsregeln werden beim Einfügen oder Aktualisieren von Amazon DocumentDB

DocumentDB-Dokumenten angewendet. Das folgende Codebeispiel zeigt Validierungsregeln für eine Sammlung von Mitarbeitern:

```
db.createCollection("employees", {
   "validator": {
      "$jsonSchema": {
         "bsonType": "object",
         "title": "employee validation",
         "required": [ "name", "employeeId"],
         "properties": {
            "name": {
                  "bsonType": "object",
                  "properties": {
                     "firstName": {
                        "bsonType": ["string"]
                     },
                     "lastName": {
                        "bsonType": ["string"]
                     }
                  },
                  "additionalProperties" : false
            },
            "employeeId": {
               "bsonType": "string",
               "description": "Unique Identifier for employee"
            },
             "salary": {
               "bsonType": "double"
            },
            "age": {
               "bsonType": "number"
            }
         },
         "additionalProperties" : true
      }
   },
   "validationLevel": "strict", "validationAction": "error"
} )
```

Ein gültiges Dokument einfügen

Im folgenden Beispiel werden Dokumente eingefügt, die den oben genannten Schemavalidierungsregeln entsprechen:

```
db.employees.insert({"name" : { "firstName" : "Carol" , "lastName" : "Smith"},
 "employeeId": "c720a" , "salary": 1000.0 })
db.employees.insert({ "name" : { "firstName" : "William", "lastName" : "Taylor" },
 "employeeId" : "c721a", "age" : 24})
```

Ein ungültiges Dokument wird eingefügt

Im folgenden Beispiel werden Dokumente eingefügt, die den obigen Schemavalidierungsregeln nicht entsprechen. In diesem Beispiel ist der EmployeeID-Wert keine Zeichenfolge:

```
db.employees.insert({
    "name" : { "firstName" : "Carol" , "lastName" : "Smith"},
    "employeeId": 720 ,
    "salary": 1000.0
})
```

Dieses Beispiel zeigt eine falsche Syntax innerhalb des Dokuments.

Eine Sammlung ändern

Der `collMod` Befehl wird verwendet, um Validierungsregeln einer vorhandenen Sammlung hinzuzufügen oder zu ändern. Im folgenden Beispiel wird der Liste der erforderlichen Felder ein Gehaltsfeld hinzugefügt:

```
db.runCommand({"collMod" : "employees",
   "validator": {
      "$jsonSchema": {
         "bsonType": "object",
         "title": "employee validation",
         "required": [ "name", "employeeId", "salary"],
         "properties": {
            "name": {
                  "bsonType": "object",
                  "properties": {
                     "firstName": {
                        "bsonType": ["string"]
                     },
                     "lastName": {
                        "bsonType": ["string"]
                     }
                  },
                  "additionalProperties" : false
```

```
            },
            "employeeId": {
               "bsonType": "string",
               "description": "Unique Identifier for employee"
            },
             "salary": {
               "bsonType": "double"
            },
            "age": {
               "bsonType": "number"
            }
         },
         "additionalProperties" : true
      }
   }
} )
```

Adressierung von Dokumenten, die vor der Änderung der Validierungsregeln hinzugefügt wurden

Verwenden Sie die folgenden `validationLevel` Modifikatoren, um Dokumente zu adressieren, die Ihrer Sammlung hinzugefügt wurden, bevor die Validierungsregeln geändert wurden:

- Strikt: Wendet Validierungsregeln auf alle Einfügungen und Aktualisierungen an.
- moderat: Wendet Validierungsregeln auf bestehende gültige Dokumente an. Bei Aktualisierungen werden vorhandene ungültige Dokumente nicht überprüft.

Im folgenden Beispiel ist nach der Aktualisierung der Validierungsregeln für die Sammlung mit dem Namen „Mitarbeiter“ das Feld Gehalt erforderlich. Die Aktualisierung des folgenden Dokuments schlägt fehl:

```
db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
```

```
"n" : 0,
    "nModified" : 0,
    "writeErrors" : [
        {
"index" : 0,
            "code" : 121,
            "errmsg" : "Document failed validation"
        }
    ],
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Wenn Sie die Validierungsstufe `moderate` auf aktualisieren, kann das obige Dokument erfolgreich aktualisiert werden:

```
db.runCommand({
    "collMod" : "employees",
    validationLevel : "moderate"
})

db.runCommand({
    update: "employees",
    updates: [{
        q: { "employeeId": "c721a" },
        u: { age: 25 , salary : 1000},
        upsert: true }]
})
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
"n" : 1,
    "nModified" : 1,
    "ok" : 1,
    "operationTime" : Timestamp(1234567890, 1)
}
```

Dokumente werden mit dem $JsonSchema abgerufen

Der `$jsonSchema` Operator kann als Filter verwendet werden, um Dokumente abzufragen, die dem JSON-Schema entsprechen. Dies ist ein Operator der obersten Ebene, der in Filterdokumenten

als Feld der obersten Ebene vorhanden sein oder mit Abfrageoperatoren wie andor, und $nor verwendet werden kann. Die folgenden Beispiele zeigen die Verwendung von $jsonSchema als Einzelfilter und mit anderen Filteroperatoren:

Dokument, das in eine Sammlung „Mitarbeiter“ eingefügt wurde:

```
{ "name" : { "firstName" : "Carol", "lastName" : "Smith" }, "employeeId" : "c720a",
 "salary" : 1000 }
{ "name" : { "firstName" : "Emily", "lastName" : "Brown" }, "employeeId" : "c720b",
 "age" : 25, "salary" : 1050.2 }
{ "name" : { "firstName" : "William", "lastName" : "Taylor" }, "employeeId" : "c721a",
 "age" : 24, "salary" : 1400.5 }
{ "name" : { "firstName" : "Jane", "lastName" : "Doe" }, "employeeId" : "c721a",
 "salary" : 1300 }
```

Sammlung wurde nur mit dem $jsonSchema Operator gefiltert:

```
db.employees.find({
       $jsonSchema: { required: ["age"] } })
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
 "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
 "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

Sammlung, die mit dem $jsonSchema Operator und einem anderen Operator gefiltert wurde:

```
db.employees.find({
       $or: [{ $jsonSchema: { required: ["age", "name"]}},
            { salary: { $lte:1000}}]});
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{ "_id" : ObjectId("64e5f8886218c620cf0e8f8a"), "name" : { "firstName" : "Carol",
 "lastName" : "Smith" }, "employeeId" : "c720a", "salary" : 1000 }
{ "_id" : ObjectId("64e5f91c6218c620cf0e8f8b"), "name" : { "firstName" : "Emily",
 "lastName" : "Brown" }, "employeeId" : "c720b", "age" : 25, "salary" : 1050.2 }
```

```
{ "_id" : ObjectId("64e5f94e6218c620cf0e8f8c"), "name" : { "firstName" : "William",
 "lastName" : "Taylor" }, "employeeId" : "c721a", "age" : 24, "salary" : 1400.5 }
```

Sammlung, die mit dem `$jsonSchema` Operator und mit dem Aggregatfilter `$match` gefiltert wurde:

```
db.employees.aggregate(
    [{ $match: {
        $jsonSchema: {
            required: ["name", "employeeId"],
            properties: {"salary" :{"bsonType": "double"}}
        }
       }
    }]
)
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
"_id" : ObjectId("64e5f8886218c620cf0e8f8a"),
 "name" : { "firstName" : "Carol", "lastName" : "Smith" },
"employeeId" : "c720a",
"salary" : 1000
}
{
"_id" : ObjectId("64e5f91c6218c620cf0e8f8b"),
"name" : { "firstName" : "Emily", "lastName" : "Brown" },
"employeeId" : "c720b",
"age" : 25,
"salary" : 1050.2
}
{
"_id" : ObjectId("64e5f94e6218c620cf0e8f8c"),
"name" : { "firstName" : "William", "lastName" : "Taylor" },
"employeeId" : "c721a",
"age" : 24,
"salary" : 1400.5
}
{
"_id" : ObjectId("64e5f9786218c620cf0e8f8d"),
"name" : { "firstName" : "Jane", "lastName" : "Doe" },
"employeeId" : "c721a",
"salary" : 1300
```

```
}
```

Bestehende Validierungsregeln anzeigen

Um die vorhandenen Validierungsregeln für eine Sammlung zu sehen, verwenden Sie:

```
db.runCommand({
    listCollections: 1,
    filter: { name: 'employees' }
})
```

Amazon DocumentDB gibt die folgende Ausgabe zurück:

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "name" : "employees",
                "type" : "collection",
                "options" : {
                    "autoIndexId" : true,
                    "capped" : false,
                    "validator" : {
                        "$jsonSchema" : {
                            "bsonType" : "object",
                            "title" : "employee validation",
                            "required" : [
                                "name",
                                "employeeId",
                                "salary"
                            ],
                            "properties" : {
                                "name" : {
                                    "bsonType" : "object",
                                    "properties" : {
                                        "firstName" : {
                                            "bsonType" : [
                                                "string"
                                            ]
                                        },
                                        "lastName" : {
                                            "bsonType" : [
```

```
                                        "string"
                                    ]
                                }
                            },
                            "additionalProperties" : false
                        },
                        "employeeId" : {
                            "bsonType" : "string",
                            "description" : "Unique Identifier for employee"
                        },
                        "salary" : {
                            "bsonType" : "double"
                        },
                        "age" : {
                            "bsonType" : "number"
                        }
                    },
                    "additionalProperties" : true
                }
            },
            "validationLevel" : "moderate",
            "validationAction" : "error"
        },
        "info" : {
            "readOnly" : false
        },
        "idIndex" : {
            "v" : 2,
            "key" : {
                "_id" : 1
            },
            "name" : "_id_",
            "ns" : "test.employees"
        }
    }
],
"id" : NumberLong(0),
"ns" : "test.$cmd.listCollections"
},
"ok" : 1,
"operationTime" : Timestamp(1692788937, 1)
}
```

Amazon DocumentDB behält auch die Validierungsregeln in der `$out` Aggregationsphase bei.

Unterstützte Schlüsselwörter

Die folgenden Felder werden in den `collMod` Befehlen `create` und unterstützt:

- **`Validator`**— Unterstützt den `$jsonSchem` Operator a.
- **`ValidationLevel`**— Unterstützt `moderate` Werte `offstrict`, und.
- **`ValidationAction`**— Unterstützt den `error` Wert.

Der $jsonSchema-Operator unterstützt die folgenden Schlüsselwörter:

- `additionalItems`
- `additionalProperties`
- `allOf`
- `anyOf`
- `bsonType`
- `dependencies`
- `description`
- `enum`
- `exclusiveMaximum`
- `exclusiveMinimum`
- `items`
- `maximum`
- `minimum`
- `maxItems`
- `minItems`
- `maxLength`
- `minLength`
- `maxProperties`
- `minProperties`
- `multipleOf`
- `not`

- `oneOf`
- `pattern`
- `patternProperties`
- `properties`
- `required`
- `title`
- `type`
- `uniqueItems`

bypassDocumentValidation

Amazon DocumentDB unterstützt `bypassDocumentValidation` die folgenden Befehle und Methoden:

- `insert`
- `update`
- `findAndModify`
- `$out`Stufe im `aggregate` Befehl und in der Methode `db.collection.aggregate()`

Amazon DocumentDB unterstützt die folgenden Befehle nicht für`bypassDocumentValidation`:

- `$merge`im `aggregate` Befehl und in der Methode `db.collection.aggregate()`
- `mapReduce`Befehl und `db.collection.mapReduce()` Methode
- `applyOps` command

Einschränkungen

Für die `$jsonSchema` Validierung gelten die folgenden Einschränkungen:

- Amazon DocumentDB gibt den Fehler „Dokument konnte nicht validiert werden“ zurück, wenn ein Vorgang die Validierungsregel nicht erfüllt.
- Elastische Amazon DocumentDB-Cluster werden nicht unterstützt`$jsonSchema`.

Amazon DocumentDB Kontingente und Beschränkungen

In diesem Thema werden die Ressourcenkontingente, Grenzwerte und Namensbeschränkungen für Amazon DocumentDB (mit MongoDB-Kompatibilität) beschrieben.

Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) und Amazon Neptune gemeinsam genutzt wird.

Themen

- Unterstützte Instance-Typen
- Unterstützte -Regionen
- Regionale Kontingente
- Aggregationsgrenzen
- Cluster-Grenzwerte
- Instance-Limits
- Benennungseinschränkungen:
- TTL-Einschränkungen
- Elastische Cluster-Grenzwerte
- Limits für elastische Cluster-Shards
- CPU-, Speicher-, Verbindungs- und Cursorlimits für Elastic Cluster pro Shard

Unterstützte Instance-Typen

Amazon DocumentDB unterstützt On-Demand-Instances und die folgenden Instance-Typen:

- NVMe-unterstützt:
  - R6GD-Instanztypen:`db.r6gd.xlarge`,,`db.r6gd.2xlarge`,`db.r6gd.4xlarge`,`db.r6gd.8xlarge`. `db.r6gd.12xlarge` `db.r6gd.16xlarge`
- Speicheroptimiert
  - R6G-Instanztypen:`db.r6g.large`,,,,`db.r6g.2xlarge`,`db.r6g.4xlarge`. `db.r6g.8xlarge` `db.r6g.12xlarge` `db.r6g.16xlarge`

  - R5-Instance-Typen:`db.r5.large`,,`db.r5.2xlarge`,`db.r5.4xlarge`,`db.r5.8xlarge`. `db.r5.12xlarge` `db.r5.16xlarge` `db.r5.24xlarge`
  - R4-Instance-Typen: `db.r4.large`, `db.r4.2xlarge`, `db.r4.4xlarge`, `db.r4.8xlarge`, `db.r4.16xlarge`.
- Überragende Leistung:
  - T4G-Instanztypen:. `db.t4g.medium`
  - T3-Instanztypen:. `db.t3.medium`

Weitere Informationen zu den unterstützten Instance-Typen und deren Spezifikationen finden Sie unter Spezifikationen der Instanzklasse.

Unterstützte -Regionen

Amazon DocumentDB ist in den folgenden AWS Regionen verfügbar:

Name der Region	Region	Verfügbarkeitszonen (Compute)
USA Ost (Ohio)	`us-east-2`	3
USA Ost (Nord-Virginia)	`us-east-1`	6
USA West (Oregon)	`us-west-2`	4
Afrika (Kapstadt)	`af-south-1`	3
Südamerika (São Paulo)	`sa-east-1`	3
Asien-Pazifik (Hongkong)	`ap-east-1`	3
Asien-Pazifik (Hyderabad)	`ap-south-2`	3
Asien-Pazifik (Mumbai)	`ap-south-1`	3
Asien-Pazifik (Seoul)	`ap-northeast-2`	4
Asien-Pazifik (Singapur)	`ap-southeast-1`	3
Asien-Pazifik (Sydney)	`ap-southeast-2`	3
Asien-Pazifik (Tokio)	`ap-northeast-1`	3
Kanada (Zentral)	`ca-central-1`	3

Name der Region	Region	Verfügbarkeitszonen (Compute)
Region China (Peking)	`cn-north-1`	3
China (Ningxia)	`cn-northwest-1`	3
Europa (Frankfurt)	`eu-central-1`	3
Europa (Irland)	`eu-west-1`	3
Europa (London)	`eu-west-2`	3
Europa (Milan)	`eu-south-1`	3
Europa (Paris)	`eu-west-3`	3
Europa (Spain)	`eu-south-2`	3
Europa (Stockholm)	`eu-north-1`	3
Naher Osten (VAE)	`me-central-1`	3
Israel (Tel Aviv)	`il-central-1`	3
AWS GovCloud (USA West)	`us-gov-west-1`	3
AWS GovCloud (US-Ost)	`us-gov-east-1`	3

Regionale Kontingente

Für bestimmte Verwaltungsfunktionen verwendet Amazon DocumentDB eine Betriebstechnologie, die mit Amazon Relational Database Service (Amazon RDS) gemeinsam genutzt wird. Die folgende Tabelle enthält regionale Beschränkungen, die von Amazon DocumentDB und Amazon RDS gemeinsam genutzt werden.

> **Note**
>
> Die oben beschriebene gemeinsame Amazon RDS-Technologie gilt nur für Amazon DocumentDB DocumentDB-Instance-basierte Cluster. Elastische Amazon DocumentDB-Cluster teilen sich keine Technologie mit Amazon RDS.

Die folgenden Beschränkungen gelten für Amazon DocumentDB DocumentDB-Instance-basierte Cluster und gelten pro AWS Konto und Region.

Ressource	AWS Standardlimit
Cluster	40
Cluster-Parametergruppen	50
Ereignisabonnements	20
Instances	40
Manuelle Cluster-Snapshots	100
Read Replicas pro Cluster	15
Subnetzgruppen	50
Subnetze pro Subnetzgruppe	20
Tags pro Ressource	50
VPC-Sicherheitsgruppen pro Instance	5

Die folgenden Beschränkungen gelten für elastische Amazon DocumentDB-Cluster und gelten pro AWS Konto und Region.

Ressource	AWS Standardlimit
Elastische Cluster	20

Ressource	AWS Standardlimit
Elastische Cluster vCPU	1024
Manueller elastischer Cluster-Snapshot	20

Sie können mit Service Quotas eine Erhöhung für ein Kontingent beantragen, sofern das Kontingent anpassbar ist. Einige Anfragen werden automatisch gelöst, während andere an gesendet werden Support. Sie können den Status eines Antrags auf Erhöhung des Kontingents verfolgen, der an eingereicht wurde Support. Anfragen zur Erhöhung der Servicekontingente erhalten keinen bevorzugten Support. Wenn Sie eine dringende Anfrage haben, wenden Sie sich bitte an Support. Weitere Informationen zu Service Quotas finden Sie unter What Is Service Quotas?.

Um eine Erhöhung des Kontingents für Amazon DocumentDB zu beantragen:

1. Öffnen Sie die Service Quotas-Konsole unter https://console.aws.amazon.com/servicequotas und melden Sie sich bei Bedarf an.
2. Wählen Sie im Navigationsbereich AWS -Services.
3. Wählen Sie Amazon DocumentDB (mit MongoDB-Kompatibilität) oder Amazon DocumentDB Elastic Cluster aus der Liste aus, oder geben Sie eines davon in das Suchfeld ein.
4. Wenn das Kontingent einstellbar ist, können Sie sein Optionsfeld oder seinen Namen auswählen und dann rechts oben auf der Seite die Option Request quota increase (Kontingenterhöhung beantragen) auswählen.
5. Geben Sie unter Change quota value (Kontingentwert ändern) den neuen Wert ein. Der neue Wert muss größer als der aktuelle Wert sein.
6. Wählen Sie Request (Anfrage). Nachdem die Anfrage genehmigt wurde, wird Applied quota value (Angewandter Kontingentwert) für das Kontingent auf den neuen Wert eingestellt.
7. Um ausstehende oder kürzlich genehmigte Anfragen anzuzeigen, wählen Sie im Navigationsbereich die Option Dashboard . Wählen Sie für ausstehende Anfragen den Status der Anfrage, um die Anfrage zu öffnen. Der Anfangsstatus einer Anfrage ist `Pending`. Nachdem sich der Status auf geändert hat`Quota requested`, sehen Sie die Fallnummer mit. Support Wählen Sie die Fallnummer, um das Ticket für Ihre Anfrage zu öffnen.

Aggregationsgrenzen

In der folgenden Tabelle werden die Aggregationsbeschränkungen in Amazon DocumentDB beschrieben.

Ressource	Limit
Maximale Anzahl der unterstützten Stages	500

Cluster-Grenzwerte

In der folgenden Tabelle werden die auf Amazon DocumentDB DocumentDB-Instances basierenden Clustergrenzwerte beschrieben.

Ressource	Limit
Cluster-Größe (Summe aller Sammlungen und Indizes)	128 TiB
Sammlungsgröße (Summe aller Sammlungen darf die Cluster-Beschränkung nicht überschreiten) – beinhaltet nicht die Indexgröße.	32 TiB
Sammlungen pro Cluster	100 000
Datenbanken pro Cluster	100 000
Datenbankgröße (Summe aller Datenbanken darf die Cluster-B eschränkung nicht überschreiten)	128 TiB
Verschachtelungstiefe für Dokumente	200 Stufen
Dokumentengröße	16 MiB
Indexschlüsselgröße	2048 Bytes
Indizes pro Sammlung	64
Schlüssel in einem zusammengesetzten Index	32

Ressource	Limit
Maximale Anzahl der Schreibvorgänge in einem einzelnen Stapelbefehl	100 000
Anzahl Benutzer pro Cluster	1000

Instance-Limits

In der folgenden Tabelle werden die Amazon DocumentDB DocumentDB-Grenzwerte pro Instance beschrieben.

Instance-Typ	Instanzspeicher (GiB)	Verbindungen (alle)	Cursor-Limit	Transaktionen öffnen	Verbindungen (aktiv)
T3. Medium	4	1000	30	50	102
T4G. Mittel	4	1000	30	50	102
R4. Groß	15,25	1700	450	N/A	1100
R4.x groß	30,5	3400	450	N/A	2700
R 4.2 x groß	61	6800	450	N/A	4500
R 4.4 x groß	122	13600	725	N/A	4500
R 4.8 x groß	288	27200	1450	N/A	4500
R 4.16 x groß	488	30000	2900	N/A	4500
R5. Groß	16	3400	450	200	1100
R5.x groß	32	7000	450	400	2700
R5.2 x groß	64	14200	450	800	4500
R 5.4 x groß	128	28400	760	1600	4500
R 5,8 x groß	256	60000	1520	3200	4500
R 5.12 x groß	384	60000	2280	4800	4500
R5.16x groß	512	60000	3040	6400	4500
R 5.24 x groß	768	60000	4560	9600	4500
R6G. Groß*	16	3400	450	200	1100

Instance-Typ	Instanzspeicher (GiB)	Verbindungen (alle)	Cursor-Limit	Transaktionen öffnen	Verbindungen (aktiv)
R6G.X groß*	32	7000	450	400	2700
R6G.2 x groß*	64	14200	450	800	4500
R6G.4 x groß*	128	28400	760	1600	4500
R6G.8 x groß*	256	60000	1520	3200	4500
R6G. 12 x groß*	384	60000	2280	4800	4500
R6G. 16 x groß*	512	60000	3040	6400	4500

* einschließlich R6GD

Mithilfe der folgenden CloudWatch Messwerte können Sie die Limits pro Instanz überwachen und Alarme auslösen. Weitere Informationen zu Amazon CloudWatch DocumentDB-Metriken finden Sie unterÜberwachen von Amazon DocumentDB mit CloudWatch.

Limit	CloudWatch Metriken
Instanzspeicher	FreeableMemory
Verbindungen	DatabaseConnectionsMax
Cursor	DatabaseCursorsMax
Transaktionen	TransactionsOpenMax

Benennungseinschränkungen:

In der folgenden Tabelle werden Benennungseinschränkungen in Amazon DocumentDB beschrieben.

Ressource	Standardlimit
Cluster Identifier (Cluster-Kennung)	• Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestri ch enden oder zwei aufeinand erfolgende Bindestriche enthalten. • Muss für alle Cluster (in Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.
Name der Sammlung: `<col>`	Die Länge beträgt [1—57] Zeichen.
Datenbankname: `<db>`	Die Länge beträgt [1—63] Zeichen.
Vollqualifizierter Sammlungsname: `<db>.<col>`	Die Länge beträgt [3—120] Zeichen.
Vollqualifizierter Indexname: `<db>.<col>$<index>`	Die Länge beträgt [6—377] Zeichen.
Indexname: `<col>$<index>`	Die Länge beträgt [3—255] Zeichen.
Instance-ID	• Die Länge beträgt [1—63] Buchstaben, Zahlen oder Bindestriche

Ressource	Standardlimit
	• Muss mit einem Buchstaben beginnen • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten • Muss für alle Instances (in Amazon RDS, Amazon Neptune und Amazon DocumentDB) pro AWS Konto und Region eindeutig sein.

Primäres Passwort	• Länge beträgt [8–100] druckbare ASCII-Zeichen.
	• Es können alle druckbaren ASCII-Zeichen mit Ausnahme der folgenden verwendet werden:
	  • **/** (Schrägstrich)
	  • **"** (doppeltes Anführungszeichen)
	  • **@** ('At'-Symbol)

Primärer Nutzername	• Die Länge beträgt [1 bis 63] alphanumerische Zeichen. • Muss mit einem Buchstaben beginnen. • Darf kein Wort sein, das von der Datenbank-Engine reserviert ist.

Ressource	Standardlimit
Parametergruppenname	• Die Länge muss [1 bis 255] alphanumerische Zeichen betragen. • Muss mit einem Buchstaben beginnen. • Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

TTL-Einschränkungen

Es kann nicht garantiert werden, dass Löschoperationen über einem TTL-Index innerhalb eines bestimmten Zeitraums abgeschlossen und mit höchster Priorität behandelt werden. Faktoren wie Ressourcenauslastung der Instance, Dokumentgröße und Gesamtdurchsatz können Einfluss auf die Dauer von TTL-Löschoperationen haben.

Elastische Cluster-Grenzwerte

In der folgenden Tabelle werden die Höchstgrenzen in elastischen Amazon DocumentDB-Clustern beschrieben.

Ressource	Limit
Elastische Cluster pro Region	20
vCPU summiert über alle elastischen Cluster pro Region	1024
Manuelle Cluster-Snapshots pro Region	20
Shards pro Cluster	32
Speicher pro Cluster (wenn die Daten gleichmäßig per Shard-Key verteilt werden)	4 PiB

Ressource	Limit
Verbindungen zum Cluster	Der niedrigere Wert von entweder 300.000 oder die Anzahl der Shards x das Verbindungslimit, das mit vCPU pro Shard verknüpft ist
UnSharded Größe der Sammlung	32 TiB
Größe der Datenerfassung (wenn die Daten gleichmäßig per Shard-Key verteilt werden)	1 PB
Datenbanken pro Cluster	10.000
UnSharded Sammlungen pro Cluster	100 000
Geteilte Sammlungen pro Cluster	1000
Benutzer pro Cluster	100
Schreibt in einem einzigen Batch-Befehl	100 000
Indizes pro Sammlung	64
Verschachtelungstiefe für Dokumente	100 Ebenen
Dokumentengröße	16 MB
Indexschlüsselgröße	2048 Bytes
Schlüssel in einem zusammengesetzten Index	32

Limits für elastische Cluster-Shards

In der folgenden Tabelle werden die maximalen Shard-Limits in elastischen Amazon DocumentDB-Clustern beschrieben.

Ressource	Limit
vCPU pro Shard-Instanz	64

Ressource	Limit
Instanzen pro Shard	16
Speicher pro Shard	128 TiB
Speicherplatz pro Sammlung pro Shard	32 TiB

CPU-, Speicher-, Verbindungs- und Cursorlimits für Elastic Cluster pro Shard

In der folgenden Tabelle werden die maximalen CPU-, Arbeitsspeicher-, Verbindungs- und Cursorlimits in Amazon DocumentDB Elastic Cluster-Shards beschrieben.

v pro CPUs Shard	Instanzspeicher (GiB)	Verbindungslimit	Cursorlimit
2	16	1700	450
4	32	3500	450
8	64	7100	450
16	128	14200	760
32	256	28400	1520
48	384	30000	2280
64	512	30000	3040

Abfragen in Amazon DocumentDB

In diesem Abschnitt werden alle Aspekte der Abfrage mit Amazon DocumentDB erklärt.

Themen

- Dokumente abfragen
- Abfrageplan
- Erläutern Sie die Ergebnisse
- Geodaten mit Amazon DocumentDB abfragen
- Teilweiser Index
- Textsuche mit Amazon DocumentDB durchführen

Dokumente abfragen

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

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

In den folgenden Codebeispielen werden die vier Dokumente verwendet, die Sie in den beiden vorangegangenen Übungen in `insertMany()` die `example` Sammlung eingefügt haben. Sie befinden sich im Abschnitt Dokumente hinzufügen von Arbeiten mit Dokumenten. `insertOne()`

Themen

- Alle Dokumente in einer Sammlung werden abgerufen
- Dokumente werden abgerufen, die einem Feldwert entsprechen
- Dokumente werden abgerufen, die einem eingebetteten Dokument entsprechen
- Dokumente werden abgerufen, die einem Feldwert in einem eingebetteten Dokument entsprechen

- Dokumente werden abgerufen, die einem Array entsprechen
- Dokumente werden abgerufen, die einem Wert in einem Array entsprechen
- Dokumente mithilfe von Operatoren abrufen

Alle Dokumente in einer Sammlung werden abgerufen

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

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

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

Dokumente werden abgerufen, die einem Feldwert entsprechen

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

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

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

Dokumente werden abgerufen, die einem eingebetteten Dokument entsprechen

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

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

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

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

Dokumente werden abgerufen, die einem Feldwert in einem eingebetteten Dokument entsprechen

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

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

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

Dokumente werden abgerufen, die einem Array entsprechen

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

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

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

Dokumente werden abgerufen, die einem Wert in einem Array entsprechen

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

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

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

Dokumente mithilfe von Operatoren abrufen

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

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

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

Abfrageplan

Wie kann ich die **`executionStats`** für einen Abfrageplan anzeigen?

Wenn Sie ermitteln, warum eine Abfrage langsamer als erwartet ausgeführt wird, kann es hilfreich sein zu verstehen, was die `executionStats` für den Abfrageplan sind. Die `executionStats` geben die Anzahl der Dokumente an, die von einer bestimmten Stufe zurückgegeben wurden (`nReturned`), die in jeder Stufe verbrachte Ausführungszeit (`executionTimeMillisEstimate`) und die Zeit, die zum Generieren eines Abfrageplans benötigt wird (`planningTimeMillis`). Sie können die zeitintensivsten Stufen Ihrer Abfrage bestimmen, um Ihre Optimierungsbemühungen aus der Ausgabe von `executionStats` zu konzentrieren, wie in den Abfragebeispielen unten gezeigt. Der `executionStats`-Parameter unterstützt derzeit keine `update`- und `delete`-Befehle.

> **Note**
>
> Amazon DocumentDB emuliert die MongoDB 3.6-API auf einer speziell entwickelten Datenbank-Engine, die ein verteiltes, fehlertolerantes, selbstheilendes Speichersystem verwendet. Daher `explain()` können sich die Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterscheiden. Kunden, die die Kontrolle über ihren Abfrageplan wünschen, können den `$hint`-Operator verwenden, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Führen Sie die Abfrage, die Sie verbessern möchten, unter dem Befehl `explain()` wie folgt aus.

```
db.runCommand({explain: {query document}}).
explain("executionStats").executionStats;
```

Im Folgenden finden Sie eine Beispieloperation.

```
db.fish.find({}).limit(2).explain("executionStats");
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "test.fish",
        "winningPlan" : {
            "stage" : "SUBSCAN",
            "inputStage" : {
                "stage" : "LIMIT_SKIP",
                "inputStage" : {
                    "stage" : "COLLSCAN"
                }
            }
        }
    },
    "executionStats" : {
        "executionSuccess" : true,
        "executionTimeMillis" : "0.063",
        "planningTimeMillis" : "0.040",
        "executionStages" : {
            "stage" : "SUBSCAN",
            "nReturned" : "2",
            "executionTimeMillisEstimate" : "0.012",
            "inputStage" : {
                "stage" : "LIMIT_SKIP",
                "nReturned" : "2",
                "executionTimeMillisEstimate" : "0.005",
                "inputStage" : {
                    "stage" : "COLLSCAN",
                    "nReturned" : "2",
                    "executionTimeMillisEstimate" : "0.005"
                }
            }
        }
```

```
    },
    "serverInfo" : {
        "host" : "enginedemo",
        "port" : 27017,
        "version" : "3.6.0"
    },
    "ok" : 1
}
```

Wenn Sie nur die `executionStats` aus der obigen Abfrage sehen möchten, können Sie den folgenden Befehl verwenden. Bei kleinen Sammlungen kann der Amazon DocumentDB DocumentDB-Abfrageprozessor entscheiden, keinen Index zu verwenden, wenn die Leistungssteigerung vernachlässigbar ist.

```
db.fish.find({}).limit(2).explain("executionStats").executionStats;
```

Cache für den Abfrageplan

Um die Leistung zu optimieren und die Planungsdauer zu verkürzen, speichert Amazon DocumentDB interne Abfragepläne im Cache. Auf diese Weise können Abfragen mit derselben Form direkt mithilfe eines zwischengespeicherten Plans ausgeführt werden.

Dieses Zwischenspeichern kann jedoch manchmal zu einer zufälligen Verzögerung für dieselbe Abfrage führen. Beispielsweise kann eine Abfrage, deren Ausführung normalerweise eine Sekunde dauert, gelegentlich zehn Sekunden dauern. Das liegt daran, dass die Leserinstanz im Laufe der Zeit verschiedene Formen der Abfrage zwischengespeichert hat, wodurch Speicherplatz verbraucht wurde. Wenn diese zufällige Verlangsamung auftritt, müssen Sie nichts unternehmen, um den Speicher freizugeben. Das System verwaltet die Speicherauslastung für Sie und sobald der Speicher einen bestimmten Schwellenwert erreicht hat, wird er automatisch freigegeben.

Erläutern Sie die Ergebnisse

Wenn Sie Informationen zu Abfrageplänen zurückgeben möchten, unterstützt Amazon DocumentDB den Ausführlichkeitsmodus. `queryPlanner` Die `explain` Ergebnisse geben den ausgewählten Abfrageplan zurück, der vom Optimierer ausgewählt wurde, in einem Format, das dem folgenden ähnelt:

```
{
```

```
   "queryPlanner" : {
      "plannerVersion" : <int>,
      "namespace" : <string>,
      "winningPlan" : {
         "stage" : <STAGE1>,
         ...
         "inputStage" : {
            "stage" : <STAGE2>,
            ...
            "inputStage" : {
               ...
            }
         }
      }
   }
}
```

In den folgenden Abschnitten werden allgemeine `explain` Ergebnisse definiert.

Themen

- Phase des Scannens und Filterns
- Schnittmenge im Index
- Vereinigung des Indexes
- Schnittpunkt/Vereinigung mehrerer Indizes
- Zusammengesetzter Index
- Phase „Sortieren“
- Gruppenphase

Phase des Scannens und Filterns

Der Optimierer kann einen der folgenden Scans wählen:

COLLSCAN

Bei dieser Phase handelt es sich um einen sequentiellen Sammlungsscan.

```
{
    "stage" : "COLLSCAN"
```

```
}
```

ISCAN

In dieser Phase werden die Indexschlüssel gescannt. Der Optimierer kann das Dokument in dieser Phase abrufen, was dazu führen kann, dass später eine FETCH-Phase angehängt wird.

```
db.foo.find({"a": 1})
{
    "stage" : "IXSCAN",
    "direction" : "forward",
    "indexName" : <idx_name>
}
```

FETCH

Wenn der Optimierer Dokumente in einer anderen Phase als IXSCAN abgerufen hat, enthält das Ergebnis eine FETCH-Phase. Die obige IXSCAN-Abfrage kann beispielsweise zu einer Kombination der Stufen FETCH und IXSCAN führen:

```
db.foo.find({"a": 1})
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXSCAN",
        "indexName" : <idx_name>
    }
}
```

IXONLYSCAN scannt nur den Indexschlüssel. Beim Erstellen zusammengesetzter Indizes wird FETCH nicht vermieden.

Schnittmenge im Index

IX UND

Amazon DocumentDB kann einen IXAND-Schritt mit einem InputStages-Array von IXSCAN enthalten, wenn es Indexüberschneidungen nutzen kann. Beispielsweise könnten wir eine Ausgabe wie die folgende sehen:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXAND",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
            }
        ]
    }
}
```

Vereinigung des Indexes

IXOR

Ähnlich wie bei der Indexüberschneidung kann Amazon DocumentDB eine `IXOR` Stufe mit einem `inputStages` Array für den `$or` Operator enthalten.

```
db.foo.find({"$or": [{"a": {"$gt": 2}}, {"b": {"$lt": 2}}]})
```

Für die obige Abfrage könnte die Ausgabe von Explain wie folgt aussehen:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                "indexName" : "a_1"
            },
            {
                "stage" : "IXSCAN",
                "indexName" : "b_1"
```

```
            }
        ]
    }
}
```

Schnittpunkt/Vereinigung mehrerer Indizes

Amazon DocumentDB kann mehrere Indexüberschneidungs- oder Union-Phasen miteinander kombinieren und dann das Ergebnis abrufen. Zum Beispiel:

```
{
    "stage" : "FETCH",
    "inputStage" : {
        "stage" : "IXOR",
        "inputStages" : [
            {
                "stage" : "IXSCAN",
                ...
            },
            {
                "stage" : "IXAND",
                "inputStages" : [
                    {
                        "stage" : "IXSCAN",
                        ...
                    },
                    {
                        "stage" : "IXSCAN",
                        ...
                    }
                ]
            }
        ]
    }
}
```

Der Indextyp (Sparse, Compound usw.) hat keinen Einfluss auf die Verwendung der Stufen „Indexüberschneidung“ oder „Union“.

Zusammengesetzter Index

Die Verwendung des zusammengesetzten Indexes von Amazon DocumentDB ist nicht auf die ersten Teilmengen indizierter Felder beschränkt. Es kann den Index mit dem Suffixteil verwenden, ist aber möglicherweise nicht sehr effizient.

Beispielsweise `{ a: 1, b: -1 }` kann der zusammengesetzte Index von alle drei folgenden Abfragen unterstützen:

```
db.orders.find( { a: 1 } )

db.orders.find( { b: 1 } )

db.orders.find( { a: 1, b: 1 } )
```

Phase „Sortieren“

Wenn es einen Index für die angeforderten Sortierschlüssel gibt, kann Amazon DocumentDB den Index verwenden, um die Reihenfolge abzurufen. In diesem Fall enthält das Ergebnis keine `SORT` Phase, sondern eine `IXSCAN` Phase. Wenn der Optimierer eine einfache Sortierung bevorzugt, wird er eine Phase wie die folgende enthalten:

```
{
    "stage" : "SORT",
    "sortPattern" : {
        "a" : 1,
        "b" : -1
    }
}
```

Gruppenphase

Amazon DocumentDB unterstützt zwei verschiedene Gruppenstrategien:

- `SORT_AGGREGATE`: Aggregat zum Sortieren auf der Festplatte.
- `HASH_AGGREGATE`: Hash-Aggregat im Speicher.

Geodaten mit Amazon DocumentDB abfragen

In diesem Abschnitt wird beschrieben, wie Sie Geodaten mit Amazon DocumentDB abfragen können. Nachdem Sie diesen Abschnitt gelesen haben, können Sie beantworten, wie Geodaten in Amazon DocumentDB gespeichert, abgefragt und indexiert werden.

Themen

- Übersicht
- Indizierung und Speicherung von Geodaten
- Abfragen von koordinatenbasierten Daten
- Einschränkungen

Übersicht

Zu den häufigsten Anwendungsfällen für Geospatial gehört die Näherungsanalyse Ihrer Daten. Zum Beispiel „Suche nach allen Flughäfen im Umkreis von 50 Meilen von Seattle“ oder „Suche nach den nächstgelegenen Restaurants an einem bestimmten Ort“. Amazon DocumentDB verwendet die GeoJSON-Spezifikation zur Darstellung von Geodaten. GeoJSON ist eine Open-Source-Spezifikation für die JSON-Formatierung von Formen in einem Koordinatenraum. GeoJSON-Koordinaten erfassen sowohl Längen- als auch Breitengrad und repräsentieren Positionen auf einer erdähnlichen Kugel.

Indizierung und Speicherung von Geodaten

Amazon DocumentDB verwendet den GeoJSON-Typ „Point“ zum Speichern von Geodaten. Jedes GeoJSON-Dokument (oder Unterdokument) besteht im Allgemeinen aus zwei Feldern:

- type — die dargestellte Form, die Amazon DocumentDB darüber informiert, wie das Feld „Koordinaten“ zu interpretieren ist. Derzeit unterstützt Amazon DocumentDB nur Punkte
- Koordinaten — ein Paar aus Breitengrad und Längengrad, dargestellt als Objekt in einem Array — [Längengrad, Breitengrad]

Amazon DocumentDB verwendet auch 2dSphere-Indizes, um Geodaten zu indizieren. Amazon DocumentDB unterstützt Indexierungspunkte. Amazon DocumentDB unterstützt Proximity-Abfragen mit 2dSphere-Indexierung.

Stellen wir uns ein Szenario vor, in dem Sie eine Anwendung für den Lieferservice von Lebensmitteln erstellen. Sie möchten die Längen- und Breitengrade verschiedener Restaurants in Amazon

DocumentDB speichern. Zu diesem Zweck empfehlen wir Ihnen, zunächst einen Index für das Geospatial-Feld zu erstellen, das das Breitengrad- und Längengradpaar enthält.

```
use restaurantsdb
db.usarestaurants.createIndex({location:"2dsphere"})
```

Die Ausgabe dieses Befehls würde etwa so aussehen:

```
{
 "createdCollectionAutomatically" : true,
 "numIndexesBefore" : 1,
 "numIndexesAfter" : 2,
 "ok" : 1
}
```

Sobald Sie einen Index erstellt haben, können Sie damit beginnen, Daten in Ihre Amazon DocumentDB-Sammlung einzufügen.

```
db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Thai Palace",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
         -122.3264,
         47.6009
      ]
   }
});

db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Noodle House",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
        -122.3517,
         47.6159
```

```
      ]
   }
});

db.usarestaurants.insert({
   "state":"Washington",
   "city":"Seattle",
   "name":"Curry House",
   "rating": 4.8,
   "location":{
      "type":"Point",
      "coordinates":[
         -121.4517,
         47.6229
      ]
   }
});
```

Abfragen von koordinatenbasierten Daten

Amazon DocumentDB unterstützt die Abfrage von räumlichen Daten nach Nähe, Inklusion und Schnittmenge. Ein gutes Beispiel für eine Näherungsabfrage ist die Suche nach allen Punkten (allen Flughäfen), die weniger als eine bestimmte Entfernung und mehr als eine Entfernung von einem anderen Punkt (Stadt) liegen. Ein gutes Beispiel für Inklusionsabfragen ist die Suche nach allen Punkten (allen Flughäfen), die sich in einem bestimmten Gebiet/Polygon (Bundesstaat New York) befinden. Ein gutes Beispiel für eine Kreuzungsabfrage ist die Suche nach einem Polygon (Bundesstaat), das sich mit einem Punkt (Stadt) überschneidet. Sie können die folgenden Geospatial-Operatoren verwenden, um Erkenntnisse aus Ihren Daten zu gewinnen.

- **`$nearSphere`**- `$nearSphere` ist ein Suchoperator, der das Suchen von Punkten unterstützt, die einem GeoJSON-Punkt am nächsten und am weitesten entfernt sind.
- **`$geoNear`**- `$geoNear` ist ein Aggregationsoperator, der die Berechnung der Entfernung von einem GeoJSON-Punkt in Metern unterstützt.
- **`$minDistance`**- `$minDistance` ist ein Suchoperator, der in Verbindung mit `$nearSphere` oder `$geoNear` zum Filtern von Dokumenten verwendet wird, die sich mindestens in der angegebenen Mindestentfernung vom Mittelpunkt befinden.
- **`$maxDistance`**- `$maxDistance` ist ein Suchoperator, der in Verbindung mit `$nearSphere` oder verwendet wird, `$geoNear` um Dokumente zu filtern, die sich höchstens in der angegebenen maximalen Entfernung vom Mittelpunkt befinden.

- **`$geoWithin`**- `$geoWithin` ist ein Suchoperator, der das Auffinden von Dokumenten mit Geodaten unterstützt, die vollständig innerhalb einer bestimmten Form, z. B. eines Polygons, vorliegen.
- **`$geoIntersects`**- `$geoIntersects` ist ein Suchoperator, der die Suche nach Dokumenten unterstützt, deren Geodaten sich mit einem angegebenen GeoJSON-Objekt überschneiden.

> **Note**
>
> `$geoNear`und `$nearSphere` benötigen einen 2dSphere-Index für das GeoJSON-Feld, das Sie in Ihrer Näherungsabfrage verwenden.

Beispiel 1

In diesem Beispiel erfahren Sie, wie Sie alle Restaurants (Punkte) nach der nächstgelegenen Entfernung zu einer Adresse (Punkt) sortiert finden.

Um eine solche Abfrage durchzuführen, können Sie die Entfernung einer Reihe von Punkten `$geoNear` zu einem anderen Punkt berechnen. Sie können auch das hinzufügen`distanceMultiplier`, um die Entfernung in Kilometern zu messen.

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001
      }
   }
])
```

Mit dem obigen Befehl würden die Restaurants sortiert nach der Entfernung (am nächsten zum weitesten) vom angegebenen Punkt zurückgegeben. Die Ausgabe dieses Befehls würde ungefähr so aussehen

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "state" : "Washington", "city" :
 "Seattle", "name" : "Noodle House", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -122.3517, 47.6159 ] }, "DistanceKilometers" : 0.03422834547294996 }
{ "_id" : ObjectId("611f3da185009a81ad38e74a"), "state" : "Washington", "city" :
 "Seattle", "name" : "Thai Palace", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -122.3264, 47.6009 ] }, "DistanceKilometers" : 2.5009390081704277 }
{ "_id" : ObjectId("611f3dae85009a81ad38e74c"), "state" : "Washington", "city" :
 "Seattle", "name" : "Curry House", "rating" : 4.8, "location" : { "type" : "Point",
 "coordinates" : [ -121.4517, 47.6229 ] }, "DistanceKilometers" : 67.52845344856914 }
```

Um die Anzahl der Ergebnisse in einer Abfrage zu begrenzen, verwenden Sie die num Option `limit` oder.

`limit`:

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001,
         "limit": 10
      }
   }
])
```

num:

```
db.usarestaurants.aggregate([
   {
      "$geoNear":{
```

```
         "near":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
         "spherical":true,
         "distanceField":"DistanceKilometers",
         "distanceMultiplier":0.001,
         "num": 10
      }
   }
])
```

> **Note**
>
> `$geoNear`stage unterstützt die num Optionen `limit` und, um die maximale Anzahl zurückzugebender Dokumente anzugeben. `$geoNear`gibt standardmäßig maximal 100 Dokumente zurück, wenn die num Optionen `limit` oder nicht angegeben sind. Dies wird durch den Wert der `$limit` Stufe außer Kraft gesetzt, falls vorhanden und der Wert kleiner als 100 ist.

Beispiel 2

In diesem Beispiel erfahren Sie, wie Sie alle Restaurants (Punkte) im Umkreis von 2 Kilometern von einer bestimmten Adresse (Punkt) finden. Um eine solche Abfrage durchzuführen, können Sie `$nearSphere` innerhalb eines Minimums `$minDistance` und Maximums `$maxDistance` von einem GeoJSON-Punkt aus

```
db.usarestaurants.find({
   "location":{
      "$nearSphere":{
         "$geometry":{
            "type":"Point",
            "coordinates":[
               -122.3516,
               47.6156
            ]
         },
```

```
         "$minDistance":1,
         "$maxDistance":2000
      }
   }
},
{
   "name":1
})
```

Mit dem obigen Befehl würden Restaurants in einer maximalen Entfernung von 2 Kilometern vom angegebenen Punkt zurückgegeben. Die Ausgabe dieses Befehls würde ungefähr so aussehen

```
{ "_id" : ObjectId("611f3da985009a81ad38e74b"), "name" : "Noodle House" }
```

Einschränkungen

Amazon DocumentDB unterstützt nicht die Abfrage oder Indizierung von Polygonen,,, LineString, MultiPoint und. MultiPolygon MultiLineString GeometryCollection

Teilweiser Index

Ein partieller Index indiziert Dokumente in einer Sammlung, die ein bestimmtes Filterkriterium erfüllen. Die Funktion für den teilweisen Index wird in instanzbasierten Amazon DocumentDB 5.0-Clustern unterstützt.

Themen

- Erstellen Sie einen Teilindex
- Unterstützte Operatoren
- Abfrage mit einem Teilindex
- Funktionen des partiellen Indexes
- Einschränkungen des partiellen Indexes

Erstellen Sie einen Teilindex

Um einen Teilindex zu erstellen, verwenden Sie die `createIndex()` Methode mit der `partialFilterExpression` Option. Mit der folgenden Operation wird beispielsweise ein

eindeutiger zusammengesetzter Index in der Orders-Auflistung erstellt, der Dokumente indexiert, die ein `OrderID` und das `isDelivered` Feld den Wert true haben:

```
db.orders.createIndex(
  {"category": 1, "CustomerId": 1, "OrderId": 1},
  {"unique": true, "partialFilterExpression":
    {"$and": [
      {"OrderId": {"$exists": true}},
      {"isDelivered": {"$eq": false}}
    ]}
  }
)
```

Unterstützte Operatoren

- $eq
- $exists
- $and (nur auf oberster Ebene)
- $gt/$gte/$lt/$lte (Indexscan wird nur verwendet, wenn der in der Abfrage angegebene Filter exakt mit dem partiellen Filterausdruck übereinstimmt) (siehe Einschränkungen)

Abfrage mit einem Teilindex

Die folgenden Abfragemuster sind mit partiellen Indizes möglich:

- Das Abfrageprädikat entspricht exakt dem Filterausdruck für den partiellen Index:

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}}
  ]}).explain()
```

- Das erwartete Ergebnis des Abfragefilters ist eine logische Teilmenge des Teilfilters:

```
db.orders.find({"$and": [
    {"OrderId": {"$exists": true}},
    {"isDelivered": {"$eq": false}},
    {"OrderAmount": {"$eq": "5"}}
```

```
  ]}).explain()
```

- Ein Unterprädikat der Abfrage kann in Verbindung mit anderen Indizes verwendet werden:

```
db.orders.createIndex({"anotherIndex":1})
db.orders.find({ "$or": [
      {"$and": [
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]},
      {"anotherIndex": {"$eq": 5}}
    ]
  }).explain()
```

> **Note**
>
> Ein Abfrageplaner kann sich dafür entscheiden, einen Sammlungsscan anstelle eines Indexscans zu verwenden, wenn dies effizient ist. Dies ist in der Regel bei sehr kleinen Sammlungen oder Abfragen der Fall, bei denen ein großer Teil einer Sammlung zurückgegeben würde.

Funktionen des partiellen Indexes

Listet Teilindizes auf

Listet Teilindizes auf, die die partialFilterExpression `getIndex` Operation verwenden. Beispielsweise listet die in ausgegebene `getIndex` Operation Teilindizes mit den Feldern key, name und partialFilterExpressions auf:

```
db.orders.getIndexes()
```

Dieses Beispiel gibt die folgende Ausgabe zurück:

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
```

```
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
  },
  {
    "v" : 4,
    "unique" : true,
    "key" : {
      "category" : 1,
      "" : 1,
      "CustomerId" : 1,
      "OrderId" : 1
    },
    "name" : "category_1_CustID_1_OrderId_1",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {
      "$and" : [
        {"OrderId": {"$exists": true}},
        {"isDelivered": {"$eq": false}}
      ]
    }
  }
]
```

Mehrere partielle Filterausdrücke für denselben Schlüssel:Reihenfolge

Für dieselben Feldkombinationen (Schlüssel:Reihenfolge) können unterschiedliche Teilindizes erstellt werden. Diese Indizes müssen einen anderen Namen haben.

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"firstPartialIndex",
    partialFilterExpression:{"OrderId":{"$exists": true}}
  }
)
```

```
db.orders.createIndex(
  {"OrderId":1},
  {
    name:"secondPartialIndex",
    partialFilterExpression:{"OrderId":{"$gt": 1000}}
  }
```

```
)
```

Führen Sie `getIndexes` den Vorgang aus, um alle Indizes in der Sammlung aufzulisten:

```
db.orders.getIndexes()
```

Diese Beispiele geben die folgende Ausgabe zurück:

```
[
  {
    "v" : 4,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "ecommerceApp.orders"
  },
  {
    "v" : 4,
    "key" : {
      "OrderId" : 1
    },
    "name" : "firstPartialIndex",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {"OrderId":{"$exists": true}}
  },
  {
    "v" : 4,
    "key" : {
      "OrderId" : 1
    },
    "name" : "secondPartialIndex",
    "ns" : "ecommerceApp.orders",
    "partialFilterExpression" : {"OrderId":{"$gt": 1000}}
  }
]
```

Important

Indexnamen müssen unterschiedlich sein und dürfen nur nach Namen gelöscht werden.

Indizes mit partiellen Eigenschaften und TTL-Eigenschaften

Sie können auch Indizes mit partiellen Eigenschaften und TTL-Eigenschaften erstellen, indem Sie bei der Indexerstellung beide `partialFilterExpression` `expireAfterSeconds` Optionen angeben. Auf diese Weise können Sie besser kontrollieren, welche Dokumente jetzt aus einer Sammlung entfernt werden.

Beispielsweise verfügen Sie möglicherweise über einen TTL-Index, der Dokumente identifiziert, die nach einem bestimmten Zeitraum gelöscht werden sollen. Mit der Option „Teilindex“ können Sie jetzt zusätzliche Bedingungen dafür festlegen, wann Dokumente gelöscht werden sollen:

```
db.orders.createIndex(
    { "OrderTimestamp": 1 },
    {
        expireAfterSeconds: 3600 ,
        partialFilterExpression: { "isDelivered": { $eq: true } }
    }
)
```

Dieses Beispiel gibt die folgende Ausgabe zurück:

```
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1,
        "operationTime" : Timestamp(1234567890, 1)
}
```

Führen Sie den `getIndexes` Vorgang aus, um die in der Sammlung vorhandenen Indizes aufzulisten:

```
db.orders.getIndexes()
[
    {
        "v" : 4,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_",
        "ns" : "test.orders"
```

```
    }
```

Dieses Beispiel gibt die folgende Ausgabe zurück:

```
[
    {
        "v": 4,
        "key": {
            "_id": 1
        },
        "name": "_id_",
        "ns": "ecommerceApp.orders"
    },
    {
        "v": 4,
        "key": {
            "OrderTimestamp": 1
        },
        "name": "OrderTimestamp_1",
        "ns": "ecommerceApp.orders",
        "partialFilterExpression": {
            "isDelivered": {
                "$eq": true
            }
        },
        "expireAfterSeconds": 3600
    }
]
```

Einschränkungen des partiellen Indexes

Für die Funktion „Teilindex“ gelten die folgenden Einschränkungen:

- Ungleichheitsabfragen in Amazon DocumentDB verwenden nur dann einen Teilindex, wenn das Abfragefilterprädikat genau dem entspricht `partialFilterExpression` und denselben Datentyp hat.

> **Note**
>
> `$hint`Kann im obigen Fall sogar nicht verwendet werden, um IXSCAN zu erzwingen.

Im folgenden Beispiel `partialFilterExpression` wird das nur angewendet, `field1` aber nicht`field2`:

```
db.orders.createIndex(
  {"OrderAmount": 1},
  {"partialFilterExpression": { OrderAmount : {"$gt" : 5}}}
)

db.orders.find({OrderAmount : {"$gt" : 5}}) // Will use partial index
db.orders.find({OrderAmount : {"$gt" : 6}}) // Will not use partial index
db.orders.find({OrderAmount : {"$gt" : Decimal128(5.00)}}) // Will not use partial
 index
```

- A `partialFilterExpression` mit Array-Operatoren werden nicht unterstützt. Der folgende Vorgang generiert einen Fehler:

```
db.orders.createIndex(
  {"CustomerId":1},
  {'partialFilterExpression': {'OrderId': {'$eq': [1000, 1001, 1002]}}}
)
```

- Die folgenden Operatoren werden im partialFilterExpression Feld nicht unterstützt:
  - `$all`(Array-Operator)
  - $mod(Array-Operator)
  - `$or`
  - `$xor`
  - `$not`
  - `$nor`
- Der Datentyp des Filterausdrucks und des Filters sollten identisch sein.

Textsuche mit Amazon DocumentDB durchführen

Mit der nativen Volltextsuchfunktion von Amazon DocumentDB können Sie mithilfe von speziellen Textindizes eine Textsuche in großen Textdatensätzen durchführen. In diesem Abschnitt werden die Funktionen der Textindex-Funktion beschrieben und Schritte zur Erstellung und Verwendung

von Textindizes in Amazon DocumentDB bereitgestellt. Einschränkungen bei der Textsuche sind ebenfalls aufgeführt.

Themen

- Unterstützte Funktionen
- Verwenden des Amazon DocumentDB-Textindex
- Unterschiede zu MongoDB
- Bewährte Verfahren und Richtlinien
- Einschränkungen

Unterstützte Funktionen

Die Amazon DocumentDB-Textsuche unterstützt die folgenden MongoDB-API-kompatiblen Funktionen:

- Erstellen Sie Textindizes für ein einzelnes Feld.
- Erstellen Sie zusammengesetzte Textindizes, die mehr als ein Textfeld enthalten.
- Führen Sie Suchen mit einem Wort oder mehreren Wörtern durch.
- Kontrollieren Sie die Suchergebnisse mithilfe von Gewichtungen.
- Sortieren Sie die Suchergebnisse nach Punktezahl.
- Verwenden Sie den Textindex in der Aggregationspipeline.
- Suchen Sie nach einer exakten Phrase.

Verwenden des Amazon DocumentDB-Textindex

Um einen Textindex für ein Feld zu erstellen, das Zeichenkettendaten enthält, geben Sie die Zeichenfolge „Text“ wie unten dargestellt an:

Index für ein einzelnes Feld:

```
db.test.createIndex({"comments": "text"})
```

Dieser Index unterstützt Textsuchabfragen im Zeichenfolgenfeld „Kommentare“ in der angegebenen Sammlung.

Erstellen Sie einen zusammengesetzten Textindex für mehr als ein Zeichenkettenfeld:

```
db.test.createIndex({"comments": "text", "title":"text"})
```

Dieser Index unterstützt Textsuchabfragen in den Zeichenfolgenfeldern „Kommentare“ und „Titel“ in der angegebenen Sammlung. Sie können bis zu 30 Felder angeben, wenn Sie einen zusammengesetzten Textindex erstellen. Nach der Erstellung werden mit Ihren Textsuchabfragen alle indizierten Felder abgefragt.

> **Note**
>
> Für jede Sammlung ist nur ein Textindex zulässig.

Einen Textindex in einer Amazon DocumentDB-Sammlung auflisten

Sie können ihn für Ihre Sammlung verwenden`getIndexes()`, um Indizes, einschließlich Textindizes, zu identifizieren und zu beschreiben, wie im folgenden Beispiel gezeigt:

```
rs0:PRIMARY> db.test.getIndexes()
[
   {
      "v" : 4,
      "key" : {
         "_id" : 1
      },
      "name" : "_id_",
      "ns" : "test.test"
   },
   {
      "v" : 1,
      "key" : {
         "_fts" : "text",
         "_ftsx" : 1
      },
      "name" : "contents_text",
      "ns" : "test.test",
      "default_language" : "english",
      "weights" : {
         "comments" : 1
      },
```

```
      "textIndexVersion" : 1
   }
]
```

Sobald Sie einen Index erstellt haben, beginnen Sie mit dem Einfügen von Daten in Ihre Amazon DocumentDB-Sammlung.

```
db.test.insertMany([{"_id": 1, "star_rating": 4, "comments": "apple is red"},
                    {"_id": 2, "star_rating": 5, "comments": "pie is delicious"},
                    {"_id": 3, "star_rating": 3, "comments": "apples, oranges - healthy
 fruit"},
                    {"_id": 4, "star_rating": 2, "comments": "bake the apple pie in the
 oven"},
                    {"_id": 5, "star_rating": 5, "comments": "interesting couch"},
                    {"_id": 6, "star_rating": 5, "comments": "interested in couch for
 sale, year 2022"}])
```

Textsuchabfragen ausführen

Führen Sie eine Textsuchabfrage mit einem Wort aus

Sie müssen die `$search` Operatoren `$text` und verwenden, um Textsuchen durchzuführen. Im folgenden Beispiel werden alle Dokumente zurückgegeben, in denen Ihr indiziertes Textfeld die Zeichenfolge „apple“ oder „apple“ in anderen Formaten wie „apples“ enthält:

```
db.test.find({$text: {$search: "apple"}})
```

Ausgabe:

Die Ausgabe dieses Befehls sieht ungefähr so aus:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Führen Sie eine Textsuche mit mehreren Wörtern durch

Sie können auch Textsuchen mit mehreren Wörtern in Ihren Amazon DocumentDB DocumentDB-Daten durchführen. Der folgende Befehl gibt Dokumente zurück, deren indiziertes Textfeld „Apfel“ oder „Kuchen“ enthält:

```
db.test.find({$text: {$search: "apple pie"}})
```

Ausgabe:

Die Ausgabe dieses Befehls sieht ungefähr so aus:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Führen Sie eine Textsuche mit mehreren Wörtern durch

Verwenden Sie für eine Phrasensuche mit mehreren Wörtern dieses Beispiel:

```
db.test.find({$text: {$search: "\"apple pie\""}})
```

Ausgabe:

Der obige Befehl gibt Dokumente zurück, deren indiziertes Textfeld genau den Ausdruck „Apfelkuchen“ enthält. Die Ausgabe dieses Befehls sieht ungefähr so aus:

```
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
```

Führen Sie eine Textsuche mit Filtern aus

Sie können die Textsuche auch mit anderen Abfrageoperatoren kombinieren, um Ergebnisse nach zusätzlichen Kriterien zu filtern:

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "interest"}}]})
```

Ausgabe:

Der obige Befehl gibt Dokumente mit einem indizierten Textfeld zurück, das jede Form von „Interesse“ und ein „Star-Rating“ gleich 5 enthält. Die Ausgabe dieses Befehls sieht ungefähr so aus:

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
{ "_id" : 6, "star_rating" : 5, "comments" : "interested in couch for sale, year
 2022" }
```

Beschränken Sie die Anzahl der Dokumente, die bei einer Textsuche zurückgegeben werden

Sie können die Anzahl der zurückgegebenen Dokumente einschränken, indem Sie Folgendes verwenden`limit`:

```
db.test.find({$and: [{star_rating: 5}, {$text: {$search: "couch"}}]}).limit(1)
```

Ausgabe:

Der obige Befehl gibt ein Ergebnis zurück, das den Kriterien des Filters entspricht:

```
{ "_id" : 5, "star_rating" : 5, "comments" : "interesting couch" }
```

Sortiert die Ergebnisse nach der Textpunktzahl

Im folgenden Beispiel werden die Ergebnisse der Textsuche nach der Textpunktzahl sortiert:

```
db.test.find({$text: {$search: "apple"}}, {score: {$meta: "textScore"}}).sort({score:
 {$meta: "textScore"}})
```

Ausgabe:

Der obige Befehl gibt Dokumente zurück, deren indiziertes Textfeld „apple“ oder „apple“ in anderen Formaten wie „apples“ enthält, und sortiert das Ergebnis danach, wie relevant das Dokument für den Suchbegriff ist. Die Ausgabe dieses Befehls sieht ungefähr so aus:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red", "score" :
 0.6079270860936958 }
{ "_id" : 3, "star_rating" : 3, "comments" : "apples, oranges - healthy fruit",
 "score" : 0.6079270860936958 }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven",
 "score" : 0.6079270860936958 }
```

`$text`und `$search` werden auch für `delete` Befehle `aggregate`,`count`,`findAndModify`,`update`, und unterstützt.

Aggregationsoperatoren

Aggregationspipeline mit **`$match`**

```
db.test.aggregate(
```

```
   [{ $match: { $text: { $search: "apple pie" } } }]
)
```

Ausgabe:

Der obige Befehl gibt die folgenden Ergebnisse zurück:

```
{ "_id" : 1, "star_rating" : 4, "comments" : "apple is red" }
{ "_id" : 3, "star_rating" : 3, "comments" : "apple - a healthy fruit" }
{ "_id" : 4, "star_rating" : 2, "comments" : "bake the apple pie in the oven" }
{ "_id" : 2, "star_rating" : 5, "comments" : "pie is delicious" }
```

Eine Kombination anderer Aggregationsoperatoren

```
db.test.aggregate(
   [
      { $match: { $text: { $search: "apple pie" } } },
      { $sort: { score: { $meta: "textScore" } } },
      { $project: { score: { $meta: "textScore" } } }
   ]
)
```

Ausgabe:

Der obige Befehl gibt die folgenden Ergebnisse zurück:

```
{ "_id" : 4, "score" : 0.6079270860936958 }
{ "_id" : 1, "score" : 0.3039635430468479 }
{ "_id" : 2, "score" : 0.3039635430468479 }
{ "_id" : 3, "score" : 0.3039635430468479 }
```

Geben Sie mehrere Felder an, wenn Sie einen Textindex erstellen

Sie können bis zu drei Feldern in Ihrem Verbundtextindex Gewichtungen zuweisen. Die Standardgewichtung, die einem Feld in einem Textindex zugewiesen wird, ist eins (1). Das Gewicht ist ein optionaler Parameter und muss im Bereich von 1 bis 100000 liegen.

```
db.test.createIndex(
   {
     "firstname": "text",
```

```
    "lastname": "text",
    ...
  },
  {
    weights: {
      "firstname": 5,
      "lastname":10,
      ...
    },
    name: "name_text_index"
  }
)
```

Unterschiede zu MongoDB

Die Textindex-Funktion von Amazon DocumentDB verwendet einen umgekehrten Index mit einem Term-Frequenz-Algorithmus. Textindizes sind standardmäßig dünn gesät. Aufgrund von Unterschieden in der Parsing-Logik, Tokenisierungstrennzeichen und anderen kann es sein, dass dieselbe Ergebnismenge wie MongoDB nicht für denselben Datensatz oder dieselbe Abfrageform zurückgegeben wird.

Es bestehen die folgenden zusätzlichen Unterschiede zwischen dem Amazon DocumentDB-Textindex und MongoDB:

- Zusammengesetzte Indizes, die Nicht-Text-Indizes verwenden, werden nicht unterstützt.
- Amazon DocumentDB-Textindizes unterscheiden nicht zwischen Groß- und Kleinschreibung.
- Für den Textindex wird nur die englische Sprache unterstützt.
- Die Textindizierung von Array-Feldern (oder Feldern mit mehreren Schlüsseln) wird nicht unterstützt. Wenn Sie beispielsweise einen Textindex für „a“ mit dem Dokument {„a“: [„apple“, „pie“]} erstellen, schlägt fehl.
- Die Indexierung von Platzhaltertext wird nicht unterstützt.
- Eindeutige Textindizes werden nicht unterstützt.
- Das Ausschließen eines Begriffs wird nicht unterstützt.

Bewährte Verfahren und Richtlinien

- Für eine optimale Leistung bei Textsuchabfragen, bei denen nach Textwerten sortiert wird, wird empfohlen, den Textindex vor dem Laden von Daten zu erstellen.

- Textindizes benötigen zusätzlichen Speicherplatz für eine optimierte interne Kopie der indizierten Daten. Dies hat zusätzliche Kosten zur Folge.

Einschränkungen

Für die Textsuche gelten in Amazon DocumentDB die folgenden Einschränkungen:

- Die Textsuche wird nur auf Amazon DocumentDB 5.0-Clustern auf Instanzbasis unterstützt.
- Textindizes speichern Lexeme und ihre Positionsinformationen. Die Gesamtgröße aller Lexeme und ihrer Positionsinformationen in einem einzigen Dokument ist auf 1 MB begrenzt.

Problembehebung bei Amazon DocumentDB

Die folgenden Abschnitte enthalten Informationen zur Behebung von Problemen, die bei der Verwendung von Amazon DocumentDB (mit MongoDB-Kompatibilität) auftreten können.

Themen

- Fehlerbehebung bei Verbindungsproblemen
- Problembehandlung bei der Indexerstellung
- Problembehebung bei Leistung und Ressourcennutzung

Fehlerbehebung bei Verbindungsproblemen

Haben Sie Probleme beim Herstellen einer Verbindung? Im Folgenden finden Sie einige gängige Szenarien und wie Sie sie lösen können.

Topics

- Es kann keine Verbindung zu einem Amazon DocumentDB DocumentDB-Endpunkt hergestellt werden
- Testen einer Verbindung zu einer Amazon DocumentDB DocumentDB-Instance
- Verbindung zu einem ungültigen Endpunkt herstellen
- Die Treiberkonfiguration wirkt sich auf die Anzahl der Verbindungen aus

Es kann keine Verbindung zu einem Amazon DocumentDB DocumentDB-Endpunkt hergestellt werden

Wenn Sie versuchen, eine Verbindung zu Amazon DocumentDB herzustellen, wird die folgende Fehlermeldung am häufigsten angezeigt.

```
connecting to: mongodb://docdb-2018-11-08-21-47-27.cluster-ccuszbx3pn5e.us-east-
1.docdb.amazonaws.com:27017/
2018-11-14T14:33:46.451-0800 W NETWORK [thread1] Failed to connect to
172.31.91.193:27017 after 5000ms milliseconds, giving up.
2018-11-14T14:33:46.452-0800 E QUERY [thread1] Error: couldn't connect to server
docdb-2018-11-08-21-47-27.cluster-ccuszbx3pn5e.us-east-1.docdb.amazonaws.com:27017,
connection attempt failed :
```

```
connect@src/mongo/shell/mongo.js:237:13
@(connect):1:6
exception: connect failed
```

Diese Fehlermeldung bedeutet in der Regel, dass Ihr Client (in diesem Beispiel die Mongo-Shell) nicht auf den Amazon DocumentDB DocumentDB-Endpunkt zugreifen kann. Dies kann aus verschiedenen Gründen der Fall sein:

Themen

- Verbindung von öffentlichen Endpunkten aus herstellen
- Regionsübergreifende Verbindungen
- Verbindung von einem anderen Amazon aus herstellen VPCs
- Die Sicherheitsgruppe blockiert eingehende Verbindungen
- Problem mit den Leseeinstellungen des Java-Mongo-Treibers

Verbindung von öffentlichen Endpunkten aus herstellen

Sie versuchen, direkt von Ihrem Laptop oder Ihrem lokalen Entwicklungscomputer aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen.

Der Versuch, direkt von einem öffentlichen Endpunkt aus, z. B. Ihrem Laptop oder Ihrem lokalen Entwicklungscomputer, eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, schlägt fehl. Amazon DocumentDB ist nur für Virtual Private Cloud (VPC) verfügbar und unterstützt derzeit keine öffentlichen Endpunkte. Daher können Sie von Ihrem Laptop oder Ihrer lokalen Entwicklungsumgebung außerhalb Ihrer VPC keine direkte Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Um von außerhalb einer Amazon VPC eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, können Sie einen SSH-Tunnel verwenden. Weitere Informationen finden Sie unter Von außerhalb einer Amazon VPC eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen. Wenn sich Ihre Entwicklungsumgebung in einer anderen Amazon VPC befindet, können Sie außerdem VPC Peering verwenden und von einer anderen Amazon VPC in derselben Region oder einer anderen Region aus eine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen.

Regionsübergreifende Verbindungen

Sie versuchen, eine Verbindung zu einem Amazon DocumentDB-Cluster in einer anderen Region herzustellen.

Wenn Sie versuchen, von einer EC2 Amazon-Instance in einer anderen Region als der Region des Clusters aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, z. B. wenn Sie versuchen, von der Region USA West (Oregon) (us-west-2) aus eine Verbindung zu einem Cluster in der Region USA Ost (Nord-Virginia) (us-east-1) herzustellen, schlägt die Verbindung fehl.

Führen Sie den folgenden Befehl aus, um die Region Ihres Amazon DocumentDB-Clusters zu überprüfen. Die Region ist im Endpunkt.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].Endpoint'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    "sample-cluster.node.us-east-1.docdb.amazonaws.com"
]
```

Führen Sie den folgenden Befehl aus, um die Region Ihrer EC2 Instance zu überprüfen.

```
 aws ec2 describe-instances \
     --query 'Reservations[*].Instances[*].Placement.AvailabilityZone'
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
[
    [
        "us-east-1a"
    ]
]
```

Verbindung von einem anderen Amazon aus herstellen VPCs

Sie versuchen, von einer VPC aus eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, die sich von der Amazon VPC unterscheidet, auf der Ihr Cluster bereitgestellt ist.

Wenn sich sowohl Ihr Amazon DocumentDB-Cluster als auch Ihre EC2 Amazon-Instance in derselben AWS-Region, aber nicht in derselben Amazon VPC befinden, können Sie keine direkte Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen, es sei denn, VPC Peering ist zwischen den beiden Amazon aktiviert. VPCs

Führen Sie den folgenden Befehl aus, um die Amazon VPC Ihrer Amazon DocumentDB-Instance zu verifizieren.

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

Führen Sie den folgenden Befehl aus, um die Amazon VPC Ihrer EC2 Amazon-Instance zu verifizieren.

```
aws ec2 describe-instances \
   --query 'Reservations[*].Instances[*].VpcId'
```

Die Sicherheitsgruppe blockiert eingehende Verbindungen

Sie versuchen, eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, und die Sicherheitsgruppe des Clusters erlaubt keine eingehenden Verbindungen auf dem Port des Clusters (Standardport: 27017).

Angenommen, Ihr Amazon DocumentDB-Cluster und Ihre EC2 Amazon-Instance befinden sich beide in derselben Region und Amazon VPC und verwenden dieselbe Amazon VPC-Sicherheitsgruppe. Wenn Sie keine Verbindung zu Ihrem Amazon DocumentDB-Cluster herstellen können, liegt das wahrscheinlich daran, dass Ihre Sicherheitsgruppe (d. h. die Firewall) für Ihren Cluster keine eingehenden Verbindungen auf dem Port zulässt, den Sie für Ihren Amazon DocumentDB-Cluster ausgewählt haben (Standardport ist 27017).

Führen Sie den folgenden Befehl aus, um den Port für Ihren Amazon DocumentDB-Cluster zu überprüfen.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[DBClusterIdentifier,Port]'
```

Führen Sie den folgenden Befehl aus, um Ihre Amazon DocumentDB-Sicherheitsgruppe für Ihren Cluster abzurufen.

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
```

```
    --query 'DBClusters[*].[VpcSecurityGroups[*],VpcSecurityGroupId]'
```

Um die Regeln für eingehende Nachrichten für Ihre Sicherheitsgruppe zu überprüfen, lesen Sie die folgenden Themen in der EC2 Amazon-Dokumentation:

- Autorisieren von eingehendem Datenverkehr für Ihre Linux-Instances
- Autorisieren von eingehendem Datenverkehr für Ihre Windows-Instances

Problem mit den Leseeinstellungen des Java-Mongo-Treibers

Die Leseeinstellungen der Clients werden nicht berücksichtigt und einige Clients können nach einem Failover nicht in Amazon DocumentDB schreiben, es sei denn, sie starten neu.

Dieses Problem, das erstmals in Java Mongo Driver 3.7.x entdeckt wurde, tritt auf, wenn ein Client mithilfe `MongoClientSettings` und insbesondere beim Verketten der Methode eine Verbindung zu Amazon DocumentDB herstellt. `applyToClusterSettings` Die MongoClient Cluster-Einstellungen können mit verschiedenen Methoden definiert werden, z. B. mit, und. `hosts()` `requiredReplicaSetName()` `mode()`

Wenn der Client in der `hosts()` Methode nur einen Host angibt, wird der Modus auf `ClusterConnectionMode.SINGLE` statt auf gesetzt. `ClusterConnectionMode.MULTIPLE` Dadurch ignoriert der Client die Lesepräferenz und stellt nur eine Verbindung zu dem Server her, der in `hosts()` konfiguriert ist. Selbst wenn die Client-Einstellungen wie unten initialisiert würden, würden alle Lesevorgänge immer noch an den primären statt an den sekundären weitergeleitet.

```
final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
    final MongoCredential credential = MongoCredential.createCredential("xxx",
            "admin", "xxxx".toCharArray());
    final MongoClientSettings settings = MongoClientSettings.builder()
            .credential(credential)
            .readPreference(ReadPreference.secondaryPreferred())
            .retryWrites(false)
            .applyToSslSettings(builder -> builder
                    .enabled(false))
            .applyToClusterSettings(builder -> builder.hosts(
                            Arrays.asList(serverAddress0
                            ))
                    .requiredReplicaSetName("rs0"))
            .build();
    MongoClient mongoClient = MongoClients.create(settings);
```

Failover-Fall

Bei Verwendung der oben genannten Client-Verbindungseinstellungen würde der Client bei einem Failover und einer verzögerten Aktualisierung des DNS-Eintrags für den Cluster-Writer-Endpunkt trotzdem versuchen, Schreibvorgänge auf den alten Writer (jetzt Reader nach dem Failover) auszuführen. Dies führt zu einem serverseitigen Fehler (nicht primär), der vom Java-Treiber nicht angemessen behandelt wird (dieser Fehler wird noch untersucht). Somit kann der Client in einem schlechten Zustand belassen werden, bis beispielsweise der Anwendungsserver neu gestartet wird.

Dafür gibt es zwei Problemumgehungen:

- Bei Clients, die über eine Verbindungszeichenfolge eine Verbindung zu Amazon DocumentDB herstellen, tritt dieses Problem nicht auf, da `ClusterConnectionMode` es `MULTIPLE` bei der Einstellung der Leseeinstellungen auf gesetzt wird.

  ```
  MongoClientURI mongoClientURI = new MongoClientURI("mongodb://usr:pass:cluster-
  endpoint:27317/test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred");
  MongoClient mongoClient = MongoClients.create(mongoClientURI.getURI());
  ```

  Oder Sie verwenden `MongoClientSettings` Builder mit der `applyConnectionString` Methode.

  ```
  final MongoClientSettings settings = MongoClientSettings.builder()
          .credential(credential)
          .applyConnectionString(new ConnectionString("usr:pass:cluster-endpoint:27317/
  test?ssl=false&replicaSet=rs0&readpreference=secondaryPreferred"))
          .retryWrites(false)
          .applyToSslSettings(builder # builder
                  .enabled(false))
          .build();
  MongoClient mongoClient = MongoClients.create(settings);
  ```

- Explizit gesetzt `ClusterConnectionMode` auf`MULTIPLE`. Dies ist nur erforderlich, wenn `applyToClusterSettings` und verwendet wird`hosts().size() == 1`.

  ```
  final ServerAddress serverAddress0 = new ServerAddress("cluster-endpoint", 27317));
  final MongoCredential credential = MongoCredential.createCredential("xxx","admin",
   "xxxx".toCharArray());
  final MongoClientSettings settings = MongoClientSettings.builder()
      .credential(credential)
      .readPreference(ReadPreference.secondaryPreferred())
  ```

```
    .retryWrites(false)
    .applyToSslSettings(builder # builder
    .enabled(false))
    .applyToClusterSettings(builder # builder
                .hosts(Arrays.asList(serverAddress0))
                .requiredReplicaSetName("rs0"))
                .mode(ClusterConnectionMode.MULTIPLE))
    .build();
MongoClient mongoClient = MongoClients.create(settings);
```

Testen einer Verbindung zu einer Amazon DocumentDB DocumentDB-Instance

Sie können Ihre Verbindung zu einem Cluster mit gängigen Linux- oder Windows-Tools testen.

Testen Sie die Verbindung über einen Linux- oder Unix-Terminal, indem Sie folgendes eingeben (ersetzen Sie `cluster-endpoint` durch den Endpunkt und ersetzen Sie `port` durch den Port Ihrer Instance).

```
nc -zv cluster-endpoint port
```

Der folgende Code ist ein Beispiel für eine Beispieloperation und den Rückgabewert:

```
nc -zv docdbTest.d4c7nm7stsfc0.us-west-2.docdb.amazonaws.com 27017

Connection to docdbTest.d4c7nm7stsfc0.us-west-2.docdb.amazonaws.com 27017 port [tcp/*]
 succeeded!
```

Verbindung zu einem ungültigen Endpunkt herstellen

Wenn Sie eine Verbindung zu einem Amazon DocumentDB-Cluster herstellen und einen Cluster-Endpunkt verwenden, der nicht gültig ist, wird ein Fehler ähnlich dem folgenden angezeigt.

```
mongo --ssl \
   --host sample-cluster.node.us-east-1.docdb.amazonaws.com:27017 \
   --sslCAFile global-bundle.pem \
   --username <user-name> \
   --password <password>
```

Das Ergebnis sieht folgendermaßen aus:

```
MongoDB shell version v3.6
connecting to: mongodb://sample-cluster.node.us-east-1.docdb.amazonaws.com:27017/
2018-11-14T17:21:18.516-0800 I NETWORK [thread1] getaddrinfo("sample-cluster.node.us-
east-1.docdb.amazonaws.com") failed:
nodename nor servname provided, or not known 2018-11-14T17:21:18.537-0800 E QUERY
 [thread1] Error: couldn't initialize
connection to host sample-cluster.node.us-east-1.docdb.amazonaws.com, address is
 invalid :
connect@src/mongo/shell/mongo.js:237:13@(connect):1:6
exception: connect failed
```

Um den gültigen Endpunkt für einen Cluster abzurufen, führen Sie den folgenden Befehl aus:

```
aws docdb describe-db-clusters \
   --db-cluster-identifier sample-cluster \
   --query 'DBClusters[*].[Endpoint,Port]'
```

Um den gültigen Endpunkt für eine Instance abzurufen, führen Sie den folgenden Befehl aus:

```
aws docdb describe-db-instances \
   --db-instance-identifier sample-instance \
   --query 'DBInstances[*].[Endpoint.Address,Endpoint.Port]'
```

Weitere Informationen finden Sie unter Grundlegendes zu Amazon DocumentDB DocumentDB-Endpunkten.

Die Treiberkonfiguration wirkt sich auf die Anzahl der Verbindungen aus

Wenn Sie den Client-Treiber verwenden, um eine Verbindung zu einem Amazon DocumentDB-Cluster herzustellen, müssen Sie den `maxPoolSize` Konfigurationsparameter berücksichtigen. Die `maxPoolSize` Einstellung bestimmt die maximale Anzahl von Verbindungen, die der Client-Treiber in seinem Verbindungspool unterhält.

Problembehandlung bei der Indexerstellung

Die folgenden Themen erläutern, was zu tun ist, wenn Ihr Index oder Ihr Indexaufbau im Hintergrund fehlschlägt.

Themen

- Die Indexerstellung schlägt fehl
- Latenzprobleme und Fehler bei der Indexerstellung im Hintergrund

Die Indexerstellung schlägt fehl

Amazon DocumentDB verwendet lokalen Speicher auf einer Instance als Teil des Indexerstellungsprozesses. Sie können diese Festplattennutzung mithilfe der FreeLocalStorage CloudWatch Metrik () `CloudWatch -> Metrics -> DocDB -> Instance Metrics` überwachen. Wenn ein Indexaufbau den gesamten lokalen Festplattenspeicherplatz verbraucht und fehlschlägt, erhalten Sie eine Fehlermeldung. Wenn Sie Daten zu Amazon DocumentDB migrieren, empfehlen wir Ihnen, zuerst Indizes zu erstellen und dann die Daten einzufügen. Weitere Informationen zu Migrationsstrategien und zum Erstellen von Indizes finden Sie Migration zu Amazon DocumentDB in der Amazon DocumentDB-Dokumentation und im Blog: Migrieren Sie mithilfe der Offline-Methode von MongoDB zu Amazon DocumentDB.

Wenn bei der Erstellung von Indizes auf einem vorhandenen Cluster die Indexerstellung länger dauert als erwartet oder fehlschlägt, empfehlen wir, dass Sie die Instance hochskalieren, um den Index zu erstellen, und dann, nachdem der Index erstellt wurde, wieder herunterskalieren. Amazon DocumentDB ermöglicht Ihnen die schnelle Skalierung von Instance-Größen innerhalb von Minuten mit dem AWS Management Console oder dem AWS CLI. Weitere Informationen finden Sie unter Instanzklassen verwalten. Bei der sekundengenauen Instance-Preisgestaltung zahlen Sie auf die Sekunde genau nur für die Ressourcen, die Sie verwenden.

Latenzprobleme und Fehler bei der Indexerstellung im Hintergrund

Index-Builds im Hintergrund in Amazon DocumentDB werden erst gestartet, wenn alle Abfragen auf der primären Instance, die vor der Initiierung der Indexerstellung gestartet wurden, vollständig ausgeführt wurden. Wenn es eine lang andauernde Abfrage gibt, werden die Index-Builds im Hintergrund blockiert, bis die Abfrage abgeschlossen ist. Daher kann es länger dauern als erwartet, bis sie abgeschlossen ist. Dies gilt auch dann, wenn Sammlungen leer sind.

Index-Builds im Vordergrund zeigen nicht dasselbe Blockierungsverhalten. Stattdessen wird bei Index-Builds im Vordergrund die Sammlung exklusiv gesperrt, bis die Indexerstellung abgeschlossen ist. Um Indizes für eine leere Sammlung zu erstellen und Blockierungen bei Abfragen mit langer Laufzeit zu vermeiden, empfehlen wir daher, Index-Builds im Vordergrund zu verwenden.

> **Note**
>
> Amazon DocumentDB lässt zu, dass zu einem bestimmten Zeitpunkt nur ein Hintergrundindex für eine Sammlung erstellt wird. Wenn DDL (Data Definition Language) Operationen wie `createIndex()` oder `dropIndex()` während eines Indexaufbaus im HIntergrund in derselben Sammlung auftreten, schlägt der Aufbau des Hintergrundindex fehl.

Problembehebung bei Leistung und Ressourcennutzung

Dieser Abschnitt enthält Fragen und Lösungen für häufig auftretende Diagnoseprobleme in Amazon DocumentDB DocumentDB-Bereitstellungen. Die bereitgestellten Beispiele verwenden die mongo-Shell und gelten für eine einzelne Instance. Informationen zum Suchen nach einem Instance-Endpunkt finden Sie unter Grundlegendes zu Amazon DocumentDB DocumentDB-Endpunkten.

Themen

- Wie ermittle ich die Anzahl der Einfüge-, Aktualisierungs- und Löschvorgänge, die für meine Sammlung über die Mongo-API ausgeführt wurden?
- Wie analysiere ich die Cache-Leistung?
- Wie finde und beende ich langsame und blockierte Abfragen?
- Wie kann ich mir einen Abfrageplan ansehen und eine Abfrage optimieren?
- Wie kann ich mir einen Abfrageplan in elastischen Clustern ansehen?
- Wie liste ich alle laufenden Operationen für eine Instance auf?
- Woher weiß ich, wann eine Abfrage Fortschritte macht?
- Wie stelle ich fest, warum ein System plötzlich langsam läuft?
- Wie ermittle ich die Ursache für eine hohe CPU-Auslastung auf einer oder mehreren Cluster-Instances?
- Wie ermittle ich die offenen Cursor auf einer Instance?
- Wie finde ich die aktuelle Version der Amazon DocumentDB DocumentDB-Engine heraus?
- Wie analysiere ich die Indexnutzung und identifiziere ungenutzte Indizes?
- Wie identifiziere ich fehlende Indizes?
- Zusammenfassung nützlicher Abfragen

Wie ermittle ich die Anzahl der Einfüge-, Aktualisierungs- und Löschvorgänge, die für meine Sammlung über die Mongo-API ausgeführt wurden?

Um die Anzahl der Einfüge-, Aktualisierungs- und Löschvorgänge anzuzeigen, die für eine bestimmte Sammlung ausgeführt wurden, führen Sie den folgenden Befehl für diese Sammlung aus:

```
db.collection.stats()
```

Die Ausgabe dieses Befehls beschreibt unter seinem `opCounters` Feld Folgendes:

- numDocsIns- Die Anzahl der Dokumente, die in diese Sammlung eingefügt wurden. Dazu gehören Dokumente, die mit den `insertMany` Befehlen `insert` und eingefügt wurden, sowie Dokumente, die mit einem Upsert eingefügt wurden.
- numDocsUpd- Die Anzahl der Dokumente, die in dieser Sammlung aktualisiert wurden. Dazu gehören Dokumente, die mit den `findAndModify` Befehlen `update` und aktualisiert wurden.
- numDocsDel- Die Anzahl der Dokumente, die aus dieser Sammlung gelöscht wurden. Dazu gehören Dokumente`deleteOne`, die mit den `findAndModify` Befehlen`deleteMany`,`remove`, und gelöscht wurden.
- LastReset — Der Zeitpunkt, zu dem diese Zähler zuletzt zurückgesetzt wurden. Die von diesem Befehl bereitgestellten Statistiken werden zurückgesetzt, wenn starting/stopping the cluster or scaling up/down die Instanz ausgeführt wird.

Ein Beispiel für die Ausgabe von running `db.collection.stats()` ist unten dargestellt.

```
{
    "ns" : "db.test",
    "count" : ...,
    "size" : ...,
    "avgObjSize" : ...,
    "storageSize" : ...,
    "capped" : false,
    "nindexes" : ...,
    "totalIndexSize" : ...,
    "indexSizes" : {
        "_id_" : ...,
        "x_1" : ...
```

```
    },
    "collScans" : ...,
    "idxScans" : ...,
    "opCounter" : {
        "numDocsIns" : ...,
        "numDocsUpd" : ...,
        "numDocsDel" : ...
    },
    "cacheStats" : {
        "collBlksHit" : ...,
        "collBlksRead" : ..,
        "collHitRatio" : ...,
        "idxBlksHit" : ...,
        "idxBlksRead" : ...,
        "idxHitRatio" : ...
    },
    "lastReset" : "2022-09-02 19:41:40.471473+00",
    "ok" : 1,
    "operationTime" : Timestamp(1662159707, 1)
}
```

Dieser Statistikbefehl sollte verwendet werden, wenn sammlungsspezifische Zähler für Einfüge-, Aktualisierungs- und Löschvorgänge über die Mongo-API angezeigt werden. Eine andere Möglichkeit, sammlungsspezifische Betriebszähler anzuzeigen, besteht darin, die DML-Überwachung zu aktivieren. Die Anzahl der Einfüge-, Aktualisierungs- und Löschvorgänge für alle Sammlungen in Zeitintervallen von einer Minute kann unter eingesehen werden. Überwachen von Amazon DocumentDB mit CloudWatch

Wie analysiere ich die Cache-Leistung?

Die Analyse der Cache-Leistung kann Aufschluss über die Effizienz des Datenabrufs und die Systemleistung geben und basiert darauf, wie viele Daten von der Festplatte im Vergleich zum Cache gelesen werden. Wir stellen Cache-Statistiken über die Anzahl der Cache-Treffer (aus dem Cache gelesene Daten) und Cache-Fehlschläge (Daten, die nicht im Cache gefunden und von der Festplatte gelesen wurden) zur Verfügung, um einen Einblick in die Cache-Leistung zu geben. Die Cache-Statistiken für eine bestimmte Sammlung können gefunden werden, indem Sie den folgenden Befehl für diese Sammlung ausführen:

```
db.collection.stats()
```

Die Werte im `cacheStats` Feld in der Ausgabe dieses Befehls liefern Cache-Statistiken für die Sammlung sowie die gesamten Cache-Statistiken für die Indizes, die für die Sammlung erstellt wurden. Diese Statistiken sind unten aufgeführt:

- **`collBlksHit`**- Die Anzahl der Blöcke, die während der Operationen an dieser Sammlung aus dem Cache gelesen wurden.
- **`collBlksRead`**- Die Anzahl der Blöcke, die bei Vorgängen an dieser Sammlung von der Festplatte gelesen wurden (Cache-Fehlschläge).
- **`collHitRatio`**— Die Cache-Trefferquote für diese Sammlung (`100 * [collBlksHit / (collBlksHit + collBlksRead)]`).
- **`idxBlksHit`**- Die Anzahl der Blöcke, die für jeden Index, der für diese Sammlung erstellt wurde, aus dem Cache gelesen wurden.
- **`idxBlksRead`**- Die Anzahl der von der Festplatte gelesenen Blöcke (Cache-Fehlschläge) für jeden Index, der für diese Sammlung erstellt wurde.
- **`idxHitRatio`**— Die Cache-Trefferquote für die für diese Sammlung erstellten Indizes (`100 * [idxBlksHit / (idxBlksHit + idxBlksRead)]`).
- **`lastReset`**- Der Zeitpunkt, zu dem diese Statistiken zuletzt zurückgesetzt wurden. Die von bereitgestellten Statistiken `db.collection.stats()` werden bei starting/stopping the cluster or scaling up/down der Instanz zurückgesetzt.

Eine Aufschlüsselung der `idxBlksRead` Felder `idxBlksHit` und für jeden Index finden Sie auch mit dem `indexStats` Befehl. Indexspezifische Cache-Statistiken finden Sie, indem Sie den folgenden Befehl ausführen:

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Für jeden Index finden Sie die folgenden Cache-Statistiken unter dem `cacheStats` Feld:

- **`blksHit`**- Die Anzahl der Blöcke, die für diesen Index aus dem Cache gelesen wurden.
- **`blksRead`**- Die Anzahl der Blöcke, die für diesen Index von der Festplatte gelesen wurden.
- **`blksHitRatio`**- Die auf vier Dezimalstellen gerundete Cache-Trefferquote, berechnet von`100 * [blksHit / (blksHit + blksRead)]`.

Wie finde und beende ich langsame und blockierte Abfragen?

Benutzerabfragen können aufgrund eines nicht optimalen Abfrageplans langsam ausgeführt oder aufgrund von Ressourcenkonflikten blockiert werden.

Zum Suchen nach Abfragen, die aufgrund eines nicht optimalen Abfrageplans verlangsamt oder aufgrund von Ressourcenkonflikten blockiert werden, verwenden Sie den Befehl `currentOp`. Sie können den Befehl filtern, um die Liste der relevanten Abfragen einzugrenzen. Mit der langsamen Abfrage muss `opid` verknüpft sein, um sie beenden zu können.

Die folgende Abfrage verwendet den Befehl `currentOp`, um alle Abfragen aufzulisten, die blockiert oder länger als 10 Sekunden ausgeführt werden.

```
db.adminCommand({
    aggregate: 1,
    pipeline: [
        {$currentOp: {}},
        {$match:
            {$or: [
                {secs_running: {$gt: 10}},
                {WaitState: {$exists: true}}]}},
        {$project: {_id:0, opid: 1, secs_running: 1}}],
    cursor: {}
});
```

Als Nächstes können Sie die Abfrage einschränken, um die `opid` von länger als 10 Sekunden ausgeführten Abfragen zu finden und diese zu beenden.

So finden und beenden Sie eine Abfrage, die länger als 10 Sekunden läuft:

1. Suchen Sie die `opid` der Abfrage.

   ```
   db.adminCommand({
       aggregate: 1,
       pipeline: [
           {$currentOp: {}},
           {$match:
               {$or:
                   [{secs_running: {$gt: 10}},
                    {WaitState: {$exists: true}}]}}],
       cursor: {}
   ```

```
});
```

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

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "opid" : 24646,
                "secs_running" : 12
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

2. Beenden Sie die Abfrage mit der Operation `killOp`.

```
db.adminCommand({killOp: 1, op: 24646});
```

Wie kann ich mir einen Abfrageplan ansehen und eine Abfrage optimieren?

Wenn eine Abfrage langsam ausgeführt wird, erfordert die Abfrageausführung möglicherweise den vollständigen Scan der Sammlung, um die relevanten Dokumente auszuwählen. Manchmal kann die Abfrage schneller ausgeführt werden, wenn geeignete Indizes erstellt werden. Mit dem Befehl `explain` erkennen Sie dieses Szenario und können die Felder für die Indizes auswählen.

> **Note**
>
> Amazon DocumentDB emuliert die MongoDB 3.6-API auf einer speziell entwickelten Datenbank-Engine, die ein verteiltes, fehlertolerantes, selbstheilendes Speichersystem verwendet. Daher `explain()` können sich die Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterscheiden. Kunden, die die Kontrolle über ihren Abfrageplan wünschen, können den `$hint`-Operator verwenden, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Führen Sie die Abfrage, die Sie verbessern möchten, unter dem Befehl `explain` wie folgt aus.

```
db.runCommand({explain: {<query document>}})
```

Im Folgenden finden Sie eine Beispieloperation.

```
db.runCommand({explain:{
    aggregate: "sample-document",
    pipeline: [{$match: {x: {$eq: 1}}}],
    cursor: {batchSize: 1}}
});
```

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

```
{
    "queryPlanner" : {
        "plannerVersion" : 1,
        "namespace" : "db.test",
        "winningPlan" : {
            "stage" : "COLLSCAN"
        }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
}
```

Die Ausgabe oben zeigt an, dass in der Phase `$match` die gesamte Sammlung gescannt werden muss, um zu prüfen, ob das Feld `"x"` in jedem Dokument gleich 1 ist. Wenn die Sammlung zahlreiche Dokumente enthält, wird der Scan der Sammlung sehr langsam ausgeführt. Damit ist auch die Gesamtleistung der Abfrage sehr niedrig. Das Vorhandensein von `"COLLSCAN"` in der Ausgabe des Befehls `explain` zeigt daher an, dass die Abfrageleistung durch die Erstellung geeigneter Indizes verbessert werden kann.

In diesem Beispiel prüft die Abfrage, ob das Feld `"x"` in allen Dokumenten gleich 1 ist. Das Erstellen eines Indexes für das Feld `"x"` ermöglicht der Abfrage, einen vollständigen Scan der Sammlung zu vermeiden und den Index zu verwenden, um die relevanten Dokumente schneller zurückzugeben.

Nach dem Erstellen eines Indexes für das Feld `"x"` sieht die `explain`-Ausgabe wie folgt aus.

```
{
    "queryPlanner" : {
         "plannerVersion" : 1,
         "namespace" : "db.test",
         "winningPlan" : {
             "stage" : "IXSCAN",
             "indexName" : "x_1",
             "direction" : "forward"
         }
    },
    "serverInfo" : {
        "host" : "...",
        "port" : ...,
        "version" : "..."
    },
    "ok" : 1
}
```

Das Erstellen eines Indexes für das Feld `"x"` ermöglicht der Stufe `$match` die Verwendung eines Index, um die Anzahl der Dokumente zu reduzieren, in denen das Prädikat `"x = 1"` ausgewertet werden muss.

Bei kleinen Sammlungen kann der Amazon DocumentDB DocumentDB-Abfrageprozessor entscheiden, keinen Index zu verwenden, wenn die Leistungssteigerung vernachlässigbar ist.

Wie kann ich mir einen Abfrageplan in elastischen Clustern ansehen?

Verwenden Sie den `explain` Befehl, um einen Abfrageplan in elastischen Clustern zu untersuchen. Im Folgenden finden Sie einen `explain` Beispielvorgang für eine Suchabfrage, die auf eine Sharded-Sammlung abzielt:

```
db.runCommand(
   {
     explain: { find: "cities", filter: {"name": "Seoul"}}
   }
)
```

> **Note**
>
> Amazon DocumentDB emuliert MongoDB auf einer speziell entwickelten Datenbank-Engine. Daher `explain()` können sich die Abfragepläne und die Ausgabe von zwischen Amazon DocumentDB und MongoDB unterscheiden. Sie können den Abfrageplan mithilfe des `$hint` Operators steuern, um die Auswahl eines bevorzugten Indexes zu erzwingen.

Die Ausgabe dieses Vorgangs könnte etwa wie folgt aussehen (JSON-Format):

```
{
  "queryPlanner" : {
    "elasticPlannerVersion" : 1,
    "winningPlan" : {
      "stage" : "SINGLE_SHARD",
      "shards" : [
        {
          "plannerVersion" : 1,
          "namespace" : "population.cities",
          "winningPlan" : {
            "stage" : "SHARD_MERGE",
            "shards" : [
              {
                "shardName" : "f2cf5cfd-fe9c-40ca-b4e5-298ca0d11111",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
                  "inputStages" : [
                    {
                      "stage" : "COLLSCAN",
                      "partitionCount" : 21
                    }
                  ]
                }
              },
              {
                "shardName" : "8f3f80e2-f96c-446e-8e9d-aab8c7f22222",
                "plannerVersion" : 1,
                "namespace" : "population.cities",
                "winningPlan" : {
                  "stage" : "PARTITION_MERGE",
```

```
                "inputStages" : [
                  {
                    "stage" : "COLLSCAN",
                    "partitionCount" : 21
                  }
                ]
              }
            },
            {
              "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a033333",
              "plannerVersion" : 1,
              "namespace" : "population.cities",
              "winningPlan" : {
                "stage" : "PARTITION_MERGE",
                "inputStages" : [
                  {
                    "stage" : "COLLSCAN",
                    "partitionCount" : 22
                  }
                ]
              }
            }
          ]
        },
        "shardName" : "32c5a06f-1b2b-4af1-8849-d7c4a0f3fb58"
      }
    ]
  }
},
"serverInfo" : {
  "host" : "example-4788267630.us-east-1.docdb-elastic.amazonaws.com:27017",
  "version" : "5.0.0"
},
"ok" : 1,
"operationTime" : Timestamp(1695097923, 1)
}
```

Die vorherige Ausgabe zeigt den Abfrageplan für die `find` Abfrage in einem Cluster mit drei Shards. Jeder Shard hat mehrere Datenpartitionen, die unterschiedliche Eingabestufen haben können. In diesem Beispiel wird ein „COLLSCAN“ (ein Sammlungsscan) auf allen Partitionen ausgeführt, bevor die Ergebnisse in der Phase“ PARTITION_MERGE „innerhalb jedes Shards

zusammengeführt werden. Die Ergebnisse aller Shards werden dann in der Phase“ SHARD_MERGE „zusammengeführt, bevor sie an den Client zurückgesendet werden.

Wie liste ich alle laufenden Operationen für eine Instance auf?

Als Benutzer oder Hauptbenutzer möchten Sie zu Diagnose- und Fehlerbehebungszwecken häufig alle aktuellen Operationen auflisten, die auf einer Instanz ausgeführt werden. (Weitere Informationen zum Verwalten von Benutzern finden Sie unter Amazon DocumentDB DocumentDB-Benutzer verwalten.)

Mit der mongo Shell können Sie die folgende Abfrage verwenden, um alle laufenden Operationen auf einer Amazon DocumentDB DocumentDB-Instance aufzulisten.

```
db.adminCommand({currentOp: 1, $all: 1});
```

Die Abfrage gibt die vollständige Liste aller Benutzerabfragen und internen Systemaufgaben zurück, die zurzeit auf der Instance ausgeführt werden.

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

```
{
    "inprog" : [
        {
            "desc" : "INTERNAL"
        },
        {
            "desc" : "TTLMonitor",
            "active" : false
        },
        {
            "client" : ...,
            "desc" : "Conn",
            "active" : true,
            "killPending" : false,
            "opid" : 195,
            "ns" : "admin.$cmd",
            "command" : {
                "currentOp" : 1,
                "$all" : 1
            },
            "op" : "command",
```

```
            "$db" : "admin",
            "secs_running" : 0,
            "microsecs_running" : NumberLong(68),
            "clientMetaData" : {
            "application" : {
                "name" : "MongoDB Shell"
            },
            "driver" : {
                ...
            },
            "os" : {
                ...
            }
         }
      },
      {
         "desc": "GARBAGE_COLLECTION",
         "garbageCollection": {
            "databaseName": "testdb",
            "collectionName": "testCollectionA"
         },
         "secs_running": 3,
         "microsecs_running": NumberLong(3123456)
      },
      {
         "desc": "GARBAGE_COLLECTION",
         "garbageCollection": {
            "databaseName": "testdb",
            "collectionName": "testCollectionB"
         },
         "secs_running": 4,
         "microsecs_running": NumberLong(4123456)
      }
   ],
   "ok" : 1
}
```

Die folgenden Werte sind gültige Werte für das Feld `"desc"`:

- **`INTERNAL`**— Interne Systemaufgaben wie die Cursorbereinigung oder Aufgaben zur Bereinigung veralteter Benutzer.
- **`TTLMonitor`**— Der Time to Live (TTL) -Monitor-Thread. Der Ausführungsstatus wird im Feld `"active"` dargestellt.

- **`GARBAGE_COLLECTION`**— Der interne Garbage-Collector-Thread.
- **`CONN`**— Die Benutzerabfrage.
- **`CURSOR`**— Die Operation ist ein inaktiver Cursor, der darauf wartet, dass der Benutzer den Befehl „GetMore“ aufruft, um den nächsten Stapel von Ergebnissen zu erhalten. In diesem Zustand verbraucht der Cursor Speicher, verbraucht aber keine Rechenleistung.

In der obigen Ausgabe werden auch alle Benutzerabfragen im System aufgeführt. Jede Benutzerabfrage wird im Kontext einer Datenbank und einer Sammlung ausgeführt. Die Kombination dieser beiden Komponenten wird als Namespace bezeichnet. Der Namespace jeder Benutzerabfrage ist im Feld `"ns"` verfügbar.

Manchmal müssen Sie alle Benutzerabfragen auflisten, die in einem bestimmten Namespace ausgeführt werden. Daher muss die vorherige Ausgabe anhand des Felds `"ns"` gefiltert werden. Im Folgenden finden Sie eine Beispielabfrage für eine gefilterte Ausgabe. Die Abfrage listet alle Benutzerabfragen auf, die zurzeit in der Datenbank `"db"` und in der Sammlung `"test"` ausgeführt werden (d. h. im Namespace `"db.test"`).

```
db.adminCommand({aggregate: 1,
    pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
               {$match: {ns: {$eq: "db.test"}}}],
    cursor: {}
});
```

Als Hauptbenutzer des Systems können Sie die Abfragen aller Benutzer und auch alle internen Systemaufgaben sehen. Alle anderen Benutzer können nur ihre jeweiligen Abfragen anzeigen.

Wenn die Gesamtzahl der Abfragen und internen Systemaufgaben die Standard-Batch-Cursor-Größe überschreitet, generiert die `mongo`-Shell automatisch das Iterator-Objekt `'it'`, um den Rest der Ergebnisse anzuzeigen. Führen Sie den Befehl `'it'` so lange aus, bis alle Ergebnisse angezeigt wurden.

Woher weiß ich, wann eine Abfrage Fortschritte macht?

Benutzerabfragen können aufgrund eines nicht optimalen Abfrageplans langsam ausgeführt oder aufgrund von Ressourcenkonflikten blockiert werden. Das Debuggen solcher Abfragen ist ein mehrstufiger Prozess, in dem möglicherweise mehrmals dieselben Schritte ausgeführt werden müssen.

Im ersten Debugging-Schritt werden alle langsamen oder blockierten Abfragen aufgeführt. Die folgende Abfrage listet alle Benutzerabfragen auf, die länger als 10 Sekunden ausgeführt wurden oder auf Ressourcen warten.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {}},
                            {$match: {$or: [{secs_running: {$gt: 10}},
                                            {WaitState: {$exists: true}}]}},
                            {$project: {_id:0,
                                        opid: 1,
                                        secs_running: 1,
                                        WaitState: 1,
                                        blockedOn: 1,
                                        command: 1}}],
                 cursor: {}
                });
```

Wiederholen Sie die vorherige Abfrage regelmäßig, um zu ermitteln, ob sich die Liste der Abfragen ändert, und die langsamen oder blockierten Abfragen zu identifizieren.

Wenn das Ausgabedokument der betreffenden Abfrage das Feld `WaitState` enthält, zeigt dies an, dass die Abfrage aufgrund von Ressourcenkonflikten langsam ist oder blockiert wird. Die Ressourcenkonflikte könnten auf E/A-Vorgänge, interne Systemaufgaben oder die Abfragen anderer Benutzer zurückgehen.

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

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "opid" : 201,
                "command" : {
                    "aggregate" : ...
                },
                "secs_running" : 208,
                "WaitState" : "IO"
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
```

```
    },
    "ok" : 1
}
```

Die E/A-Schnittstelle kann einen Engpass darstellen, wenn zahlreiche Abfragen in verschiedenen Sammlungen auf derselben Instance gleichzeitig ausgeführt werden oder die Instance zu klein für den Datensatz ist, auf dem die Abfrage ausgeführt wird. Wenn es sich bei den Abfragen um schreibgeschützte Abfragen handelt, können Sie der eben beschriebenen Situation durch Aufteilung der Abfragen für die einzelnen Sammlungen auf separate Replikate abhelfen. Wenn in verschiedenen Sammlungen gleichzeitig Updates ausgeführt werden oder die Instance zu klein für den Datensatz ist, können Sie die Instance aufwärts skalieren.

Wenn der Ressourcenkonflikt auf Abfragen anderer Benutzer zurückzuführen ist, zeigt das Feld `"blockedOn"` im Ausgabedokument die `"opid"` der Abfrage an, die sich auf die Abfrage auswirkt. Folgen Sie mithilfe der `"opid"` der Kette der Felder `"WaitState"` und `"blockedOn"` aller Abfragen, um die Abfrage an der Ende der Kette zu finden.

Wenn es sich bei der Aufgabe am Ende der Kette um eine interne Aufgabe handelt, kann das Problem nur behoben werden, indem die Abfrage beendet und später erneut ausgeführt wird.

Im Folgenden finden Sie eine Beispielausgabe, in der die Suchabfrage auf einer Sammlungssperre blockiert wird, die im Besitz einer anderen Aufgabe ist.

```
{
    "inprog" : [
        {
            "client" : "...",
            "desc" : "Conn",
            "active" : true,
            "killPending" : false,
            "opid" : 75,
            "ns" : "...",
            "command" : {
                "find" : "...",
                "filter" : {

                }
            },
            "op" : "query",
            "$db" : "test",
            "secs_running" : 9,
            "microsecs_running" : NumberLong(9449440),
```

```
            "threadId" : 24773,
            "clientMetaData" : {
                "application" : {
                   "name" : "MongoDB Shell"
                },
                "driver" : {
                    ...
                },
                "os" : {
                    ...
                }
            },
            "WaitState" : "CollectionLock",
            "blockedOn" : "INTERNAL"
        },
        {
            "desc" : "INTERNAL"
        },
        {
            "client" : "...",
            ...
            "command" : {
                "currentOp" : 1
            },
            ...
        }
    ],
    "ok" : 1
}
```

Wenn `"WaitState"` die Werte `"Latch"`, `"SystemLock"`, `"BufferLock"`, `"BackgroundActivity"` oder `"Other"` hat, wird der Ressourcenkonflikt von internen Systemaufgaben verursacht. Wenn die Situation lange andauert, besteht die einzige Abhilfe darin, die Abfrage zu beenden und später erneut auszuführen.

Wie stelle ich fest, warum ein System plötzlich langsam läuft?

Im Folgenden finden Sie einige häufige Gründe für die Verlangsamung von Systemen:

- Übermäßige Ressourcenkonflikte zwischen gleichzeitigen Abfragen
- Zunahme der Anzahl der aktiven gleichzeitigen Abfragen im Laufe der Zeit
- Interne Systemaufgaben wie `"GARBAGE_COLLECTION"`

Wenn Sie die Systemnutzung im Laufe der Zeit überwachen möchten, führen Sie in regelmäßigen Abständen die folgende `"currentOp"`-Abfrage aus, und geben Sie die Ergebnisse in einem externen Speicher aus. Die Abfrage zählt die Anzahl der Abfragen und Operationen in jedem Namespace im System. Sie können anschließend die Systemnutzungsergebnisse analysieren, um die Systemlast zu verstehen und entsprechende Maßnahmen zu ergreifen.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Diese Abfrage gibt eine Aggregation aller in den einzelnen Namespaces ausgeführten Abfragen, alle internen Systemaufgaben und die eindeutige Anzahl der Wartezustände (wenn vorhanden) pro Namespace zurück.

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

```
{
    "waitedMS" : NumberLong(0),
    "cursor" : {
        "firstBatch" : [
            {
                "_id" : {
                    "desc" : "Conn",
                    "ns" : "db.test",
                    "WaitState" : "CollectionLock"
                },
               "count" : 2
            },
            {
                "_id" : {
                    "desc" : "Conn",
                    "ns" : "admin.$cmd"
                },
                "count" : 1
            },
            {
                "_id" : {
                    "desc" : "TTLMonitor"
                },
```

```
                "count" : 1
            }
        ],
        "id" : NumberLong(0),
        "ns" : "admin.$cmd"
    },
    "ok" : 1
}
```

In der oben gezeigten Ausgabe werden zwei Benutzerabfragen im Namespace `"db.test"` durch eine Sammlungssperre blockiert: 1 Abfrage in Namespace `"admin.$cmd"` und eine interne `"TTLMonitor"`-Aufgabe.

Wenn die Ausgabe viele Abfragen mit blockierenden Wartezuständen anzeigt, finden Sie Informationen unter Wie finde und beende ich langsame und blockierte Abfragen?.

Wie ermittle ich die Ursache für eine hohe CPU-Auslastung auf einer oder mehreren Cluster-Instances?

In den folgenden Abschnitten finden Sie Informationen, die Ihnen möglicherweise helfen, die Ursache für die hohe CPU-Nutzung einer Instance zu ermitteln. Die Ergebnisse sind vom Workload abhängig.

- Informationen dazu, wie Sie ermitteln, warum eine Instance plötzlich langsam ausgeführt wird, finden Sie unter Wie stelle ich fest, warum ein System plötzlich langsam läuft?.
- Informationen zum Identifizieren und Beenden langsamer Abfragen auf einer bestimmten Instance finden Sie unter Wie finde und beende ich langsame und blockierte Abfragen?.
- Informationen dazu, wie Sie ermitteln, ob eine Abfrage fortschreitet, finden Sie unter Woher weiß ich, wann eine Abfrage Fortschritte macht?.
- Informationen dazu, warum die Ausführung einer Abfrage lange dauert, finden Sie unter Wie kann ich mir einen Abfrageplan ansehen und eine Abfrage optimieren?.
- Informationen dazu, wie Sie langsame Abfragen über die Zeit nachverfolgen, finden Sie unter Profilierung von Amazon DocumentDB DocumentDB-Vorgängen.

Abhängig vom Grund für die hohe CPU-Nutzung Ihrer Instance können Sie einen oder mehrere der folgenden Schritte ausführen.

- Wenn die primäre Instance eine hohe CPU-Nutzung aufweist, die Replikat-Instances jedoch nicht, sollten Sie den Datenverkehr für Lesevorgänge über die Client-Leseeinstellungen auf mehrere

Replikate verteilen (z. B. `secondaryPreferred`). Weitere Informationen finden Sie unter Als Replikatsatz eine Verbindung zu Amazon DocumentDB herstellen.

Die Verwendung von Replikaten für Lesevorgänge kann zu einer besseren Nutzung der Cluster-Ressourcen führen, da die primäre Instance eine größere Menge an Schreibdatenverkehr verarbeiten kann. Lesevorgänge aus -Replikaten sind Eventually Consistent.

- Wenn die hohe CPU-Nutzung durch Schreib-Workloads verursacht wird, können Sie durch die Skalierung der Cluster-Instances auf einen größeren Instance-Typ die Anzahl der CPU-Kerne erhöhen, die für den Workload verfügbar sind. Weitere Informationen erhalten Sie unter Instances und Spezifikationen der Instanzklasse.
- Wenn alle Cluster-Instances eine hohe CPU-Nutzung aufweisen und der Workload Replikate für Lesevorgänge verwendet, erhöht das Hinzufügen weiterer Replikate zum Cluster die Zahl der für Lesevorgänge verfügbaren Ressourcen. Weitere Informationen finden Sie unter Hinzufügen einer Amazon DocumentDB DocumentDB-Instance zu einem Cluster.

Wie ermittle ich die offenen Cursor auf einer Instance?

Wenn Sie mit einer Amazon DocumentDB DocumentDB-Instance verbunden sind, können Sie den Befehl verwenden, `db.runCommand("listCursors")` um die geöffneten Cursor auf dieser Instance aufzulisten. Es gibt ein Limit von bis zu 4.560 aktiven Cursorn, die zu einem bestimmten Zeitpunkt auf einer bestimmten Amazon DocumentDB DocumentDB-Instance geöffnet sind, je nach Instance-Typ. Es wird allgemein empfohlen, Cursor zu schließen, die nicht mehr verwendet werden, da Cursor Ressourcen auf einer Instance verwenden und eine Obergrenze haben. Spezifische Grenzwerte finden Sie unterAmazon DocumentDB Kontingente und Beschränkungen.

```
db.runCommand("listCursors")
```

Wie finde ich die aktuelle Version der Amazon DocumentDB DocumentDB-Engine heraus?

Führen Sie den folgenden Befehl aus, um Ihre aktuelle Amazon DocumentDB DocumentDB-Engine-Version zu ermitteln.

```
db.runCommand({getEngineVersion: 1})
```

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

```
{ "engineVersion" : "2.x.x", "ok" : 1 }
```

> **Note**
>
> Die Engine-Version für Amazon DocumentDB 3.6 ist 1.x.x und die Engine-Version für Amazon DocumentDB 4.0 ist 2.x.x.

Wie analysiere ich die Indexnutzung und identifiziere ungenutzte Indizes?

Zum Identifizieren der Indizes für eine bestimmte Sammlung führen Sie den folgenden Befehl aus:

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

Um zu analysieren, wie viele Indizes bei Vorgängen an den Sammlungen verwendet werden, können die `indexStats` Befehle `collStats` und verwendet werden. Führen Sie den folgenden Befehl aus, um die Gesamtzahl der mit Indizes (Indexscans) durchgeführten Scans mit der Anzahl der ohne Index durchgeführten Scans (Sammlungsscans) zu vergleichen:

```
db.collection.stats()
```

Die Ausgabe für diesen Befehl umfasst die folgenden Werte:

- **`idxScans`**- Die Anzahl der Scans, die für diese Sammlung mithilfe eines Indexes durchgeführt wurden.
- **`collScans`**- Die Anzahl der Scans, die für diese Sammlung ohne Verwendung eines Indexes durchgeführt wurden. Bei diesen Scans hätten die Dokumente in der Sammlung nacheinander durchsucht werden müssen.
- **`lastReset`**- Der Zeitpunkt, zu dem diese Zähler zuletzt zurückgesetzt wurden. Die von diesem Befehl bereitgestellten Statistiken werden zurückgesetzt, wenn starting/stopping the cluster or scaling up/down die Instanz ausgeführt wird.

Eine Aufschlüsselung, wie häufig die einzelnen Indizes verwendet werden, finden Sie in der Ausgabe des folgenden Befehls. Es hat sich bewährt, ungenutzte Indizes regelmäßig zu identifizieren und zu entfernen, um die Leistung zu verbessern und die Kosten zu senken, da dadurch unnötige Rechenleistung, Speicher und I/Os für die Verwaltung der Indizes vermieden werden.

```
db.collection.aggregate([{$indexStats:{}}]).pretty()
```

Die Ausgabe dieses Befehls enthält die folgenden Werte für jeden Index, der für die Sammlung erstellt wurde:

- **`ops`**- Die Anzahl der Operationen, die den Index verwendet haben. Wenn Ihre Workload über einen ausreichend langen Zeitraum ausgeführt wurde und Sie sicher sind, dass sich Ihre Workload in einem konstanten Zustand befindet, würde ein `ops`-Wert von Null anzeigen, dass der Index überhaupt nicht verwendet wird.
- **`numDocsRead`**- Die Anzahl der Dokumente, die bei Vorgängen gelesen wurden, die diesen Index verwenden.
- **`since`**— Die Zeit, seit Amazon DocumentDB mit der Erfassung von Statistiken zur Indexnutzung begonnen hat. Dies ist in der Regel der Wert seit dem letzten Datenbankneustart oder der letzten Wartungsaktion.
- **`size`**- Die Größe dieses Indexes in Byte.

Das folgende Beispiel ist eine Beispielausgabe aus der Ausführung des obigen Befehls:

```
{
    "name" : "_id_",
    "key" : {
        "_id" : 1
    },
    "host" : "example-host.com:12345",
    "size" : NumberLong(...),
    "accesses" : {
        "ops" : NumberLong(...),
        "docsRead" : NumberLong(...),
        "since" : ISODate("...")
    },
    "cacheStats" : {
        "blksRead" : NumberLong(...),
        "blksHit" : NumberLong(...),
        "hitRatio" : ...
    }
}
{
    "name" : "x_1",
    "key" : {
```

```
        "x" : 1
    },
    "host" : "example-host.com:12345",
    "size" : NumberLong(...),
    "accesses" : {
        "ops" : NumberLong(...),
        "docsRead" : NumberLong(...),
        "since" : ISODate("...")
    },
    "cacheStats" : {
        "blksRead" : NumberLong(...),
        "blksHit" : NumberLong(...),
        "hitRatio" : ...
    }
}
```

Um die Gesamtindexgröße für eine Sammlung zu ermitteln, führen Sie den folgenden Befehl aus:

```
db.collection.stats()
```

Führen Sie den folgenden Befehl aus, um einen nicht verwendeten Index zu löschen:

```
db.collection.dropIndex("indexName")
```

Wie identifiziere ich fehlende Indizes?

Sie können den Amazon DocumentDB DocumentDB-Profiler verwenden, um langsame Abfragen zu protokollieren. Eine Abfrage, die wiederholt im Protokoll über langsame Abfragen erscheint, kann darauf hinweisen, dass ein zusätzlicher Index erforderlich ist, um diese Abfrageleistung zu verbessern.

Sie können Möglichkeiten für hilfreiche Indizes identifizieren, indem Sie nach Abfragen mit langer Laufzeit suchen, die über eine oder mehrere Phasen verfügen, die mindestens eine `COLLSCAN` Phase ausführen. Das bedeutet, dass in der Abfragephase jedes Dokument in der Sammlung gelesen werden muss, um eine Antwort auf die Abfrage zu geben.

Das folgende Beispiel zeigt eine Abfrage zu einer Sammlung von Taxifahrten, die auf einer großen Sammlung ausgeführt wurden.

```
db.rides.count({"fare.totalAmount":{$gt:10.0}}))
```

Um dieses Beispiel auszuführen, musste die Abfrage einen Sammlungsscan durchführen (d. h. jedes einzelne Dokument in der Sammlung lesen), da es keinen Index für das `fare.totalAmount`-Feld gibt. Die Ausgabe des Amazon DocumentDB DocumentDB-Profilers für diese Abfrage sieht in etwa wie folgt aus:

```
{
    ...
    "cursorExhausted": true,
    "nreturned": 0,
    "responseLength": 0,
    "protocol": "op_query",
    "millis": 300679,
    "planSummary": "COLLSCAN",
    "execStats": {
        "stage": "COLLSCAN",
        "nReturned": "0",
        "executionTimeMillisEstimate": "300678.042"
    },
    "client": "172.31.5.63:53878",
    "appName": "MongoDB Shell",
    "user": "example"
}
```

Um die Abfrage in diesem Beispiel zu beschleunigen, möchten Sie einen Index auf `fare.totalAmount` erstellen, wie unten gezeigt.

```
db.rides.createIndex( {"fare.totalAmount": 1}, {background: true} )
```

> **Note**
>
> Indizes, die im Vordergrund erstellt werden (d. h. wenn die Option `{background:true}` beim Erstellen des Index nicht angegeben wurde), erhalten eine exklusive Schreibsperre, die Anwendungen daran hindert, Daten so lange in die Sammlung zu schreiben, bis der Indexaufbau abgeschlossen ist. Beachten Sie diese mögliche Auswirkung beim Erstellen von Indizes für Produktions-Cluster. Beim Erstellen von Indizes empfehlen wir die Einstellung `{background:true}`.

Im Allgemeinen möchten Sie Indizes für Felder mit hoher Kardinalität erstellen (z. B. eine große Anzahl eindeutiger Werte). Das Erstellen eines Index für ein Feld mit geringer Kardinalität kann zu

einem großen Index führen, der nicht verwendet wird. Der Amazon DocumentDB DocumentDB-Abfrageoptimierer berücksichtigt bei der Erstellung eines Abfrageplans die Gesamtgröße der Sammlung und die Selektivität der Indizes. Manchmal wird der Abfrageprozessor einen `COLLSCAN` auswählen, selbst wenn ein Index vorhanden ist. Dies geschieht, wenn der Abfrageprozessor schätzt, dass die Verwendung des Index keinen Leistungsvorteil gegenüber dem Scannen der gesamten Sammlung bringt. Wenn Sie den Abfrageprozessor zwingen möchten, einen bestimmten Index zu verwenden, können Sie den `hint()`-Operator wie unten gezeigt verwenden.

```
db.collection.find().hint("indexName")
```

Zusammenfassung nützlicher Abfragen

Die folgenden Abfragen können für die Überwachung der Leistung und der Ressourcennutzung in Amazon DocumentDB nützlich sein.

- Verwenden Sie den folgenden Befehl, um Statistiken über eine bestimmte Sammlung anzuzeigen, einschließlich Betriebszähler, Cache-Statistiken, Zugriffsstatistiken und Größenstatistiken:

  ```
  db.collection.stats()
  ```

- Verwenden Sie den folgenden Befehl, um Statistiken zu jedem Index anzuzeigen, der für eine Sammlung erstellt wurde, einschließlich der Größe des Indexes, indexspezifischer Cache-Statistiken und Statistiken zur Indexnutzung:

  ```
  db.collection.aggregate([{$indexStats:{}}]).pretty()
  ```

- Verwenden Sie die folgende Abfrage, um alle Aktivitäten aufzulisten.

  ```
  db.adminCommand({currentOp: 1, $all: 1});
  ```

- Der folgende Code listet alle langsamen oder blockierten Abfragen auf.

  ```
  db.adminCommand({aggregate: 1,
                   pipeline: [{$currentOp: {}},
                              {$match: {$or: [{secs_running: {$gt: 10}},
                                              {WaitState: {$exists: true}}]}},
                              {$project: {_id:0,
                                          opid: 1,
                                          secs_running: 1,
                                          WaitState: 1,
  ```

```
                                    blockedOn: 1,
                                    command: 1}}],
                 cursor: {}
                });
```

- Der folgende Code beendet eine Abfrage.

```
db.adminCommand({killOp: 1, op: <opid of running or blocked query>});
```

- Verwenden Sie den folgenden Code, um eine aggregierte Ansicht des Systemstatus zu erhalten.

```
db.adminCommand({aggregate: 1,
                 pipeline: [{$currentOp: {allUsers: true, idleConnections: true}},
                            {$group: {_id: {desc: "$desc", ns: "$ns", WaitState:
 "$WaitState"}, count: {$sum: 1}}}],
                 cursor: {}
                });
```

Referenz zur Amazon DocumentDB-Cluster-, Instance- und Ressourcenmanagement-API

In diesem Abschnitt werden die Cluster-, Instance- und Ressourcenverwaltungsvorgänge für Amazon DocumentDB (mit MongoDB-Kompatibilität) beschrieben, auf die über HTTP, das AWS Command Line Interface (AWS CLI) oder das AWS SDK zugegriffen werden kann. Sie können diese verwenden, APIs um Cluster und Instances zu erstellen, zu löschen und zu ändern.

> **Important**
>
> Diese APIs werden nur für die Verwaltung von Clustern, Instances und zugehörigen Ressourcen verwendet. Informationen darüber, wie Sie eine Verbindung zu einem laufenden Amazon DocumentDB-Cluster herstellen, finden Sie unterLeitfaden für die ersten Schritte.

Themen

- Aktionen
- Datentypen
- Häufige Fehler
- Geläufige Parameter

Aktionen

Die folgenden Aktionen werden unterstützt von Amazon DocumentDB (with MongoDB compatibility):

- AddSourceIdentifierToSubscription
- AddTagsToResource
- ApplyPendingMaintenanceAction
- CopyDBClusterParameterGroup
- CopyDBClusterSnapshot
- CreateDBCluster
- CreateDBClusterParameterGroup
- CreateDBClusterSnapshot

- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster
- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters
- DescribeDBClusterSnapshotAttributes
- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- FailoverGlobalCluster
- ListTagsForResource

- ModifyDBCluster
- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource
- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster
- SwitchoverGlobalCluster

Die folgenden Aktionen werden von Amazon DocumentDB Elastic Clusters unterstützt:

- ApplyPendingMaintenanceAction
- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot
- GetCluster
- GetClusterSnapshot
- GetPendingMaintenanceAction
- ListClusters
- ListClusterSnapshots

- ListPendingMaintenanceActions
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster
- TagResource
- UntagResource
- UpdateCluster

Amazon DocumentDB (with MongoDB compatibility)

Die folgenden Aktionen werden unterstützt von Amazon DocumentDB (with MongoDB compatibility):

- AddSourceIdentifierToSubscription
- AddTagsToResource
- ApplyPendingMaintenanceAction
- CopyDBClusterParameterGroup
- CopyDBClusterSnapshot
- CreateDBCluster
- CreateDBClusterParameterGroup
- CreateDBClusterSnapshot
- CreateDBInstance
- CreateDBSubnetGroup
- CreateEventSubscription
- CreateGlobalCluster
- DeleteDBCluster
- DeleteDBClusterParameterGroup
- DeleteDBClusterSnapshot
- DeleteDBInstance
- DeleteDBSubnetGroup
- DeleteEventSubscription
- DeleteGlobalCluster

- DescribeCertificates
- DescribeDBClusterParameterGroups
- DescribeDBClusterParameters
- DescribeDBClusters
- DescribeDBClusterSnapshotAttributes
- DescribeDBClusterSnapshots
- DescribeDBEngineVersions
- DescribeDBInstances
- DescribeDBSubnetGroups
- DescribeEngineDefaultClusterParameters
- DescribeEventCategories
- DescribeEvents
- DescribeEventSubscriptions
- DescribeGlobalClusters
- DescribeOrderableDBInstanceOptions
- DescribePendingMaintenanceActions
- FailoverDBCluster
- FailoverGlobalCluster
- ListTagsForResource
- ModifyDBCluster
- ModifyDBClusterParameterGroup
- ModifyDBClusterSnapshotAttribute
- ModifyDBInstance
- ModifyDBSubnetGroup
- ModifyEventSubscription
- ModifyGlobalCluster
- RebootDBInstance
- RemoveFromGlobalCluster
- RemoveSourceIdentifierFromSubscription
- RemoveTagsFromResource

- ResetDBClusterParameterGroup
- RestoreDBClusterFromSnapshot
- RestoreDBClusterToPointInTime
- StartDBCluster
- StopDBCluster
- SwitchoverGlobalCluster

AddSourceIdentifierToSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Fügt eine Quell-ID einem Abonnement für Ereignisbenachrichtigungen hinzu.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceIdentifier

Die Kennung der Ereignisquelle, die hinzugefügt werden soll:

- Wenn der Quelltyp eine Instanz ist, `DBInstanceIdentifier` muss a angegeben werden.
- Wenn der Quelltyp eine Sicherheitsgruppe ist, `DBSecurityGroupName` muss a angegeben werden.
- Wenn der Quelltyp eine Parametergruppe ist, `DBParameterGroupName` muss a angegeben werden.
- Wenn der Quelltyp ein Snapshot ist, `DBSnapshotIdentifier` muss a angegeben werden.

Typ: Zeichenfolge

Erforderlich: Ja

SubscriptionName

Der Name des Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen, zu dem Sie eine Quell-ID hinzufügen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

SourceNotFound

Die angeforderte Quelle konnte nicht gefunden werden.

HTTP Status Code: 404

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

AddTagsToResource

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Fügt einer Amazon DocumentDB DocumentDB-Ressource Metadaten-Tags hinzu. Sie können diese Tags in Kostenzuordnungsberichten verwenden, um Kosten nachzuverfolgen, die mit Amazon DocumentDB-Ressourcen verbunden sind, oder in einer `Condition` Abrechnung in einer AWS Identity and Access Management (IAM-) Richtlinie für Amazon DocumentDB.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ResourceName

Die Amazon DocumentDB DocumentDB-Ressource, zu der die Tags hinzugefügt werden. Dieser Wert ist ein Amazon-Ressourcenname.

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die der Amazon DocumentDB DocumentDB-Ressource zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ApplyPendingMaintenanceAction

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Wendet eine ausstehende Wartungsaktion auf eine Ressource an (z. B. auf eine Amazon DocumentDB DocumentDB-Instance).

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ApplyAction

Die anstehende Wartungsaktion, die auf diese Ressource angewendet werden soll.

Zulässige Werte: `system-update`, `db-upgrade`

Typ: Zeichenfolge

Erforderlich: Ja

OptInType

Ein Wert, der die Art der Opt-in-Anfrage angibt oder eine Opt-in-Anfrage rückgängig macht. Eine Opt-in-Anfrage vom Typ `immediate` kann nicht rückgängig gemacht werden.

Zulässige Werte:

- `immediate` – Die Wartungsmaßnahme sofort anwenden.
- `next-maintenance` – Die Wartungsaktion im nächsten Wartungsfenster für die Ressource anwenden.
- `undo-opt-in` – Alle bestehenden `next-maintenance`-Opt-In-Anfragen stornieren.

Typ: Zeichenfolge

Erforderlich: Ja

ResourceIdentifier

Der Amazon-Ressourcenname (ARN) der Ressource, auf die sich die anstehende Wartungsaktion bezieht.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

ResourcePendingMaintenanceActions

Stellt die Ausgabe von darApplyPendingMaintenanceAction.

Typ: ResourcePendingMaintenanceActions Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

ResourceNotFoundFault

Die angegebene Ressourcen-ID wurde nicht gefunden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CopyDBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Kopiert die angegebene Cluster-Parametergruppe.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceDBClusterParameterGroupIdentifier

Der Bezeichner oder der Amazon-Ressourcenname (ARN) für die Quell-Cluster-Parametergruppe.

Einschränkungen:

- Muss eine gültige Cluster-Parametergruppe angeben.
- Wenn sich die Quellcluster-Parametergruppe in derselben Datei AWS-Region wie die Kopie befindet, geben Sie eine gültige Parametergruppen-ID an, z. B.`my-db-cluster-param-group`, oder einen gültigen ARN.
- Wenn sich die Quellparametergruppe in einer anderen AWS-Region als der Kopie befindet, geben Sie einen gültigen Cluster-Parametergruppen-ARN an, z. B.`arn:aws:rds:us-east-1:123456789012:sample-cluster:sample-parameter-group`.

Typ: Zeichenfolge

Erforderlich: Ja

TargetDBClusterParameterGroupDescription

Eine Beschreibung für die kopierte Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

TargetDBClusterParameterGroupIdentifier

Der Bezeichner für die kopierte Cluster-Parametergruppe.

Einschränkungen:

- Kann nicht Null, leer oder negativ sein.

- Muss zwischen 1 und 255 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster-param-group1`

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die der Parametergruppe zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterParameterGroup

Detaillierte Informationen zu einer Cluster-Parametergruppe.

Typ: DBClusterParameterGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupAlreadyExists

Eine Parametergruppe mit demselben Namen ist bereits vorhanden.

HTTP Status Code: 400

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

DBParameterGroupQuotaExceeded

Diese Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Parametergruppen überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CopyDBClusterSnapshot

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Kopiert einen Snapshot eines Clusters.

Um einen Cluster-Snapshot aus einem gemeinsam genutzten manuellen Cluster-Snapshot zu kopieren, `SourceDBClusterSnapshotIdentifier` muss es sich um den Amazon-Ressourcennamen (ARN) des gemeinsam genutzten Cluster-Snapshots handeln. Sie können einen freigegebenen DB-Cluster-Snapshot, ob verschlüsselt oder nicht, nur in dieselbe AWS-Region kopieren.

Um den Kopiervorgang abzubrechen, nachdem er ausgeführt wurde, löschen Sie den Ziel-Cluster-Snapshot, der identifiziert wurde, `TargetDBClusterSnapshotIdentifier` solange sich dieser Cluster-Snapshot im Kopierstatus befindet.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceDBClusterSnapshotIdentifier

Die ID des zu kopierenden Cluster-Snapshots. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Es ist ein gültiger System-Snapshot im Status verfügbar erforderlich.
- Wenn sich der Quell-Snapshot im selben Zustand AWS-Region wie die Kopie befindet, geben Sie eine gültige Snapshot-ID an.
- Wenn sich der Quell-Snapshot in einer anderen Version AWS-Region als der Kopie befindet, geben Sie einen gültigen Cluster-Snapshot-ARN an.

Beispiel: `my-cluster-snapshot1`

Typ: Zeichenfolge

Erforderlich: Ja

TargetDBClusterSnapshotIdentifier

Die ID des neuen Cluster-Snapshots, der aus dem Quell-Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster-snapshot2`

Typ: Zeichenfolge

Erforderlich: Ja

CopyTags

Stellen Sie diese Option ein`true`, um alle Tags aus dem Quell-Cluster-Snapshot in den Ziel-Cluster-Snapshot zu kopieren, andernfalls`false`. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüssel-ID für einen verschlüsselten Cluster-Snapshot. Die AWS KMS Schlüssel-ID ist der Amazon-Ressourcenname (ARN), die AWS KMS Schlüssel-ID oder der AWS KMS Schlüsselalias für den AWS KMS Verschlüsselungsschlüssel.

Wenn Sie einen verschlüsselten Cluster-Snapshot von Ihrem kopieren AWS-Konto, können Sie einen Wert angeben, `KmsKeyId` um die Kopie mit einem neuen AWS KMS Verschlüsselungsschlüssel zu verschlüsseln. Wenn Sie keinen Wert für angeben`KmsKeyId`, wird die Kopie des Cluster-Snapshots mit demselben AWS KMS Schlüssel verschlüsselt wie der Quell-Cluster-Snapshot.

Wenn Sie einen verschlüsselten Cluster-Snapshot, der gemeinsam genutzt wird AWS-Konto, von einem anderen kopieren, müssen Sie einen Wert für angeben`KmsKeyId`.

Um einen verschlüsselten Cluster-Snapshot in einen anderen `KmsKeyId` zu kopieren AWS-Region, geben Sie die AWS KMS Schlüssel-ID ein, mit der Sie die Kopie des Cluster-Snapshots in der Zielregion verschlüsseln möchten. AWS KMS Verschlüsselungsschlüssel sind spezifisch für AWS-Region das Land, in dem sie erstellt wurden, und Sie können keine Verschlüsselungsschlüssel von einem AWS-Region in einem anderen AWS-Region verwenden.

Wenn Sie einen unverschlüsselten Cluster-Snapshot kopieren und einen Wert für den `KmsKeyId` Parameter angeben, wird ein Fehler zurückgegeben.

Typ: Zeichenfolge

Erforderlich: Nein

PreSignedUrl

Die URL, die eine mit Signature Version 4 signierte Anfrage für die `CopyDBClusterSnapshot` API-Aktion enthält AWS-Region , die den zu kopierenden Quell-Cluster-Snapshot enthält. Sie müssen den `PreSignedUrl` Parameter verwenden, wenn Sie einen Cluster-Snapshot von einem anderen kopieren AWS-Region.

Wenn Sie ein AWS SDK-Tool oder das verwenden AWS CLI, können Sie `SourceRegion` (oder `--source-region` für AWS CLI) angeben, anstatt es `PreSignedUrl` manuell anzugeben. Durch die Angabe wird `SourceRegion` automatisch eine vorsignierte URL generiert, bei der es sich um eine gültige Anforderung für den Vorgang handelt, die in der Quelle ausgeführt werden kann. AWS-Region

Bei der vorsignierten URL muss es sich um eine gültige Anforderung für die `CopyDBClusterSnapshot` API-Aktion handeln, die in der Quelle ausgeführt werden kann AWS-Region , die den zu kopierenden Cluster-Snapshot enthält. Die vorsignierte URL-Anforderung muss die folgenden Parameterwerte enthalten:

- `SourceRegion`- Die ID der Region, die den zu kopierenden Snapshot enthält.
- `SourceDBClusterSnapshotIdentifier`— Die Kennung für den verschlüsselten Cluster-Snapshot, der kopiert werden soll. Dieser Bezeichner muss im ARN-Format (Amazon-Ressourcenname) der Quell- AWS-Region angegeben werden. Wenn Sie beispielsweise einen verschlüsselten Cluster-Snapshot von us-east-1 kopieren AWS-Region, `SourceDBClusterSnapshotIdentifier` sieht Ihr etwa wie folgt aus:. `arn:aws:rds:us-east-1:12345678012:sample-cluster:sample-cluster-snapshot`
- `TargetDBClusterSnapshotIdentifier`- Der Bezeichner für den neuen Cluster-Snapshot, der erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Typ: Zeichenfolge

Erforderlich: Nein

Stichworte.Tag.N

Die Tags, die dem Cluster-Snapshot zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterSnapshot

Detaillierte Informationen zu einem Cluster-Snapshot.

Typ: DBClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotAlreadyExistsFault

Sie haben bereits einen Cluster-Snapshot mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen AWS KMS Schlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateDBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen neuen Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Cluster-ID. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster`

Typ: Zeichenfolge

Erforderlich: Ja

Engine

Der Name der Datenbank-Engine, die für diesen Cluster verwendet werden soll.

Zulässige Werte: `docdb`

Typ: Zeichenfolge

Erforderlich: Ja

AvailabilityZones. AvailabilityZoneN

Eine Liste der Amazon EC2 Availability Zones, in denen Instances im Cluster erstellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

BackupRetentionPeriod

Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Sie müssen einen Mindestwert von 1 angeben.

Standard: 1

Einschränkungen:

- Muss ein Wert zwischen 1 und 35 sein.

Typ: Ganzzahl

Erforderlich: Nein

DBClusterParameterGroupName

Der Name der Parameter-Gruppe, die mit diesem Cluster zu verknüpfen ist.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupName

Eine DB-Subnetzgruppe, die diesem DB-Cluster zugeordnet werden soll.

Einschränkungen: Der Wert muss mit dem Namen einer vorhandenen `DBSubnetGroup` übereinstimmen. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn `DeletionProtection` aktiviert, kann der Cluster nur gelöscht werden, wenn er geändert und `DeletionProtection` deaktiviert wurde. `DeletionProtection`schützt Cluster davor, versehentlich gelöscht zu werden.

Typ: Boolesch

Erforderlich: Nein

EnableCloudwatchLogsExports.Mitglied.N

Eine Liste von Protokolltypen, die für den Export nach Amazon CloudWatch Logs aktiviert werden müssen. Sie können Prüf- oder Profiler-Protokolle aktivieren. Weitere Informationen finden Sie unter Amazon DocumentDB DocumentDB-Ereignisse prüfen und Amazon DocumentDB DocumentDB-Operationen profilieren.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

EngineVersion

Die Versionsnummer der zu verwendenden Datenbank-Engine. Die `--engine-version` wird standardmäßig auf die neueste Hauptversion des Moduls festgelegt. Bei Produktions-Workloads empfehlen wir, diesen Parameter explizit mit der beabsichtigten Hauptversion zu deklarieren.

Typ: Zeichenfolge

Erforderlich: Nein

GlobalClusterIdentifier

Die Cluster-ID des neuen globalen Clusters.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüssel-ID für einen verschlüsselten Cluster.

Die AWS KMS Schlüssel-ID ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Cluster erstellen AWS-Konto , dem der AWS KMS Verschlüsselungsschlüssel gehört, der zum Verschlüsseln des neuen Clusters verwendet wird, können Sie den AWS KMS Schlüsselalias anstelle des ARN für den AWS KMS Verschlüsselungsschlüssel verwenden.

Wenn ein Verschlüsselungsschlüssel in `KmsKeyId` nicht angegeben ist:

- Wenn der `StorageEncrypted`-Parameter `true` lautet, verwendet Amazon DocumentDB Ihren Standard-Verschlüsselungsschlüssel.

AWS KMS erstellt den Standard-Verschlüsselungsschlüssel für Ihren AWS-Konto. Ihr AWS-Konto hat für jeden einen anderen Standard-Verschlüsselungsschlüssel AWS-Regionen.

Typ: Zeichenfolge

Erforderlich: Nein

ManageMasterUserPassword

Gibt an, ob das Masterbenutzerpasswort mit Amazon Web Services Secrets Manager verwaltet werden soll.

Einschränkung: Sie können das Masterbenutzer-Passwort nicht mit Amazon Web Services Secrets Manager verwalten, wenn `MasterUserPassword` es angegeben ist.

Typ: Boolesch

Erforderlich: Nein

MasterUsername

Der Name des Masterbenutzers für diesen DB-Cluster.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben oder Zahlen enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf kein Wort sein, das für die ausgewählte Datenbank-Engine reserviert ist.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserPassword

Das Passwort für den Masterbenutzer der Datenbank. Dieses Passwort kann alle druckbaren ASCII-Zeichen, außer Schrägstrich (/), doppeltes Anführungszeichen (") oder das "At"-Zeichen (@), enthalten.

Einschränkungen: Muss 8 bis 100 Zeichen enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserSecretKmsKeyId

Die Amazon Web Services KMS-Schlüssel-ID zur Verschlüsselung eines Geheimnisses, das automatisch im Amazon Web Services Secrets Manager generiert und verwaltet wird. Diese Einstellung ist nur gültig, wenn das Masterbenutzerkennwort von Amazon DocumentDB im Amazon Web Services Secrets Manager für den DB-Cluster verwaltet wird.

Die Amazon Web Services KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Um einen KMS-Schlüssel in einem anderen Amazon Web Services Services-Konto zu verwenden, geben Sie den Schlüssel-ARN oder den Alias-ARN an.

Wenn Sie nichts angeben`MasterUserSecretKmsKeyId`, wird der `aws/secretsmanager` KMS-Schlüssel zur Verschlüsselung des Geheimnisses verwendet. Wenn sich das Geheimnis in einem anderen Amazon Web Services Services-Konto befindet, können Sie den `aws/secretsmanager` KMS-Schlüssel nicht verwenden, um das Geheimnis zu verschlüsseln, und Sie müssen einen vom Kunden verwalteten KMS-Schlüssel verwenden.

Es gibt einen Standard-KMS-Schlüssel für Ihr Amazon Web Services Services-Konto. Ihr Amazon Web Services Services-Konto hat für jede Amazon Web Services Services-Region einen anderen Standard-KMS-Schlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Portnummer, auf der die Instances im Cluster Verbindungen akzeptieren.

Typ: Ganzzahl

Erforderlich: Nein

PreferredBackupWindow

Der tägliche Zeitraum, in dem automatische Sicherungen erstellt werden, wenn diese mit dem Parameter `BackupRetentionPeriod` aktiviert sind.

Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils AWS-Region 8 Stunden ausgewählt wird.

Einschränkungen:

- Muss im Format `hh24:mi-hh24:mi` angegeben werden.
- Muss in Universal Coordinated Time (UTC) angegeben werden.
- Darf nicht mit dem bevorzugten Wartungsfenster in Konflikt treten.
- Muss mindestens 30 Minuten betragen.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock ausgewählt wird und an einem zufälligen Wochentag stattfindet. AWS-Region

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

PreSignedUrl

Wird derzeit nicht unterstützt.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Gibt an, ob der Cluster verschlüsselt ist.

Typ: Boolesch

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Gültige Werte für den Speichertyp - `standard | iopt1`

Der Standardwert ist `standard`

> **Note**
>
> Wenn Sie einen DocumentDB-DB-Cluster mit dem Speichertyp auf erstellen`iopt1`, wird der Speichertyp in der Antwort zurückgegeben. Der Speichertyp wird nicht zurückgegeben, wenn Sie ihn auf `standard` setzen.

Typ: Zeichenfolge

Erforderlich: Nein

Stichworte.Tag.N

Die Tags, die dem Cluster zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

VpcSecurityGroupIds. VpcSecurityGroupIdN

Eine Liste von EC2 VPC-Sicherheitsgruppen, die diesem Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName`bezieht sich nicht auf eine bestehende Cluster-Parametergruppe.

HTTP Status Code: 404

DBClusterQuotaExceededFault

Der Cluster kann nicht erstellt werden, da Sie das maximal zulässige Kontingent an Clustern erreicht haben.

HTTP Status Code: 403

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

GlobalClusterNotFoundFault

Das bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InsufficientStorageClusterCapacity

Für die aktuelle Aktion ist nicht genügend Speicherplatz verfügbar. Möglicherweise können Sie diesen Fehler beheben, indem Sie Ihre Subnetzgruppe so aktualisieren, dass sie verschiedene Availability Zones verwendet, in denen mehr Speicherplatz verfügbar ist.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

InvalidDBSubnetGroupStateFault

Die Subnetzgruppe kann nicht gelöscht werden, da sie verwendet wird.

HTTP Status Code: 400

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, solange sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nach ihrer Erstellung nicht alle Availability Zones ab, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen AWS KMS Schlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instanzen verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python

- AWS SDK for Ruby V3

CreateDBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt eine neue Cluster-Parametergruppe.

Parameter in einer Cluster-Parametergruppe gelten für alle Instances in einem Cluster.

Zu Beginn wird eine Cluster-Parametergruppe mit den Standardparametern für die Datenbank-Engine erstellt, die von den Instances im Cluster verwendet wird. In Amazon DocumentDB können Sie keine direkten Änderungen an der `default.docdb3.6` Cluster-Parametergruppe vornehmen. Wenn Ihr Amazon DocumentDB-Cluster die Standard-Cluster-Parametergruppe verwendet und Sie einen Wert darin ändern möchten, müssen Sie zuerst eine neue Parametergruppe erstellen oder eine bestehende Parametergruppe kopieren, sie ändern und dann die geänderte Parametergruppe auf Ihren Cluster anwenden. Damit die neue Cluster-Parametergruppe und die zugehörigen Einstellungen wirksam werden, müssen Sie anschließend die Instances im Cluster ohne Failover neu starten. Weitere Informationen finden Sie unter Ändern von Amazon DocumentDB-Cluster-Parametergruppen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der Cluster-Parametergruppe.

Einschränkungen:

- Darf nicht dem Namen einer vorhandenen `DBClusterParameterGroup` entsprechen.

> **Note**
>
> Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Typ: Zeichenfolge

Erforderlich: Ja

DBParameterGroupFamily

Der Name der Cluster-Parametergruppenfamilie.

Typ: Zeichenfolge

Erforderlich: Ja

Description

Die Beschreibung der Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

Stichworte.Tag.N

Die Tags, die der Cluster-Parametergruppe zugeordnet werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterParameterGroup

Detaillierte Informationen zu einer Cluster-Parametergruppe.

Typ: DBClusterParameterGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupAlreadyExists

Eine Parametergruppe mit demselben Namen ist bereits vorhanden.

HTTP Status Code: 400

DBParameterGroupQuotaExceeded

Diese Anforderung würde dazu führen, dass Sie die zulässige Anzahl von Parametergruppen überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateDBClusterSnapshot

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen Snapshot eines Clusters.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die ID des Clusters, für den ein Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Beispiel: `my-cluster`

Typ: Zeichenfolge

Erforderlich: Ja

DBClusterSnapshotIdentifier

Die ID des Cluster-Snapshots. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster-snapshot1`

Typ: Zeichenfolge

Erforderlich: Ja

Tags.Tag.N

Die Tags, die dem Cluster-Snapshot zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterSnapshot

Detaillierte Informationen zu einem Cluster-Snapshot.

Typ: DBClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterSnapshotAlreadyExistsFault

Sie haben bereits einen Cluster-Snapshot mit der angegebenen Kennung.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateDBInstance

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erzeugt eine neue Instanz.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die ID des Clusters, zu dem die Instance gehört.

Typ: Zeichenfolge

Erforderlich: Ja

DBInstanceClass

Die Rechen- und Speicherkapazität der Instance, beispielsweise `db.r5.large`.

Typ: Zeichenfolge

Erforderlich: Ja

DBInstanceIdentifier

Die Instance-ID. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `mydbinstance`

Typ: Zeichenfolge

Erforderlich: Ja

Engine

Der Name der Datenbank-Engine, die für diese Instance verwendet werden soll.

Zulässiger Wert: `docdb`

Typ: Zeichenfolge

Erforderlich: Ja

AutoMinorVersionUpgrade

Dieser Parameter gilt nicht für Amazon DocumentDB. Amazon DocumentDB führt unabhängig vom eingestellten Wert keine kleineren Versions-Upgrades durch.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

AvailabilityZone

Die Amazon EC2 Availability Zone, in der die Instance erstellt wurde.

Standard: Eine zufällige, vom System gewählte Availability Zone auf dem Endpunkt. AWS-Region

Beispiel: `us-east-1d`

Typ: Zeichenfolge

Erforderlich: Nein

CACertificateIdentifier

Die CA-Zertifikatskennung, die für das Serverzertifikat der DB-Instance verwendet werden soll.

Weitere Informationen finden Sie unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten bei der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Typ: Zeichenfolge

Erforderlich: Nein

CopyTagsToSnapshot

Ein Wert, der angibt, ob Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden sollen. Standardmäßig werden Tags nicht kopiert.

Typ: Boolesch

Erforderlich: Nein

EnablePerformanceInsights

Ein Wert, der angibt, ob Performance Insights für die DB-Instance aktiviert werden sollen. Weitere Informationen finden Sie unter Verwenden von Amazon Performance Insights.

Typ: Boolesch

Erforderlich: Nein

PerformanceInsightsKMSKeyId

Die AWS KMS Schlüssel-ID für die Verschlüsselung von Performance Insights Insights-Daten.

Die AWS KMS Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel.

Wenn Sie keinen Wert für PerformanceInsights KMSKey Id angeben, verwendet Amazon DocumentDB Ihren Standard-KMS-Schlüssel. Es gibt einen Standard-KMS-Schlüssel für Ihr Amazon Web Services Services-Konto. Ihr Amazon Web Services Services-Konto hat für jede Amazon Web Services Services-Region einen anderen Standard-KMS-Schlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock ausgewählt wird und an einem zufälligen Wochentag stattfindet. AWS-Region

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance heraufgestuft wird.

Standard: 1

Gültige Werte: 0-15

Typ: Ganzzahl

Erforderlich: Nein

Stichworte.Tag.N

Die Tags, die der Instance zugewiesen werden sollen. Sie können einer Instance bis zu 10 Tags zuweisen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instanz.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AuthorizationNotFound

Die angegebene CIDR-IP oder EC2 Amazon-Sicherheitsgruppe ist für die angegebene Sicherheitsgruppe nicht autorisiert.

Amazon DocumentDB ist möglicherweise auch nicht autorisiert, die erforderlichen Aktionen in Ihrem Namen mithilfe von IAM durchzuführen.

HTTP Status Code: 404

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceAlreadyExists

Sie haben bereits eine Instanz mit der angegebenen ID.

HTTP Status Code: 400

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

DBSecurityGroupNotFound

`DBSecurityGroupName`bezieht sich nicht auf eine bestehende Sicherheitsgruppe.

HTTP Status Code: 404

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

InstanceQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Instanzen überschreiten.

HTTP Status Code: 400

InsufficientDBInstanceCapacity

Die angegebene Instanzklasse ist in der angegebenen Availability Zone nicht verfügbar.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nach ihrer Erstellung nicht alle Availability Zones ab, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen AWS KMS Schlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instanzen verfügbar ist.

HTTP Status Code: 400

StorageTypeNotSupported

Der angegebene Speicher `StorageType` kann der DB-Instance nicht zugeordnet werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateDBSubnetGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt eine neue Subnetzgruppe. Subnetzgruppen müssen mindestens ein Subnetz in mindestens zwei Availability Zones in der enthalten. AWS-Region

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupDescription

Die Beschreibung für die Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Ja

DBSubnetGroupName

Der Name für die Subnetzgruppe. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen: Darf nicht mehr als 255 Buchstaben, Ziffern, Punkte, Unterstriche, Leerzeichen und Bindestriche enthalten. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Ja

SubnetIds. SubnetIdentifierN

Das EC2 Amazon-Subnetz IDs für die Subnetzgruppe.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

Stichworte.Tag.N

Das Tag, das der Subnetzgruppe zugeordnet werden soll.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBSubnetGroup

Detaillierte Informationen zu einer Subnetzgruppe.

Typ: DBSubnetGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupAlreadyExists

`DBSubnetGroupName`wird bereits von einer vorhandenen Subnetzgruppe verwendet.

HTTP Status Code: 400

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Subnetzgruppen überschreiten.

HTTP Status Code: 400

DBSubnetQuotaExceededFault

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Subnetzen in einer Subnetzgruppe überschreiten.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateEventSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt ein Abonnement für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen. Für diese Aktion ist ein Thema mit dem Amazon Resource Name (ARN) erforderlich, das mithilfe der Amazon DocumentDB DocumentDB-Konsole, der Amazon SNS SNS-Konsole oder der Amazon SNS SNS-API erstellt wurde. Um einen ARN mit Amazon SNS zu erhalten, müssen Sie ein Thema in Amazon SNS erstellen und das Thema abonnieren. Der ARN wird in der Amazon SNS SNS-Konsole angezeigt.

Sie können den Typ der Quelle (`SourceType`) angeben, über die Sie benachrichtigt werden möchten. Sie können auch eine Liste der Amazon DocumentDB DocumentDB-Quellen (`SourceIds`) bereitstellen, die die Ereignisse auslösen, und Sie können eine Liste von Ereigniskategorien (`EventCategories`) für Ereignisse bereitstellen, über die Sie benachrichtigt werden möchten. Sie können beispielsweise angeben`SourceType = db-instance`, `SourceIds = mydbinstance1, mydbinstance2` und`EventCategories = Availability, Backup`.

Wenn Sie sowohl als auch angeben `SourceIds` (z. B. `SourceType = db-instance` und`SourceIdentifier = myDBInstance1`), werden Sie über alle `db-instance` Ereignisse für die angegebene Quelle informiert. `SourceType` Wenn Sie eine angeben, `SourceType` aber keine angeben`SourceIdentifier`, erhalten Sie eine Benachrichtigung über die Ereignisse für diesen Quelltyp für alle Ihre Amazon DocumentDB DocumentDB-Quellen. Wenn Sie weder das noch das `SourceType` angeben, werden Sie über Ereignisse informiert`SourceIdentifier`, die aus allen Amazon DocumentDB DocumentDB-Quellen generiert wurden, die zu Ihrem Kundenkonto gehören.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SnsTopicArn

Der Amazon-Ressourcenname (ARN) des SNS-Themas, das für die Ereignisbenachrichtigung erstellt wurde. Amazon SNS erstellt den ARN, wenn Sie ein Thema erstellen und es abonnieren.

Typ: Zeichenfolge

Erforderlich: Ja

SubscriptionName

Der Name des Abonnements.

Einschränkungen: Der Name muss weniger als 255 Zeichen lang sein.

Typ: Zeichenfolge

Erforderlich: Ja

Enabled

Ein boolescher Wert. Wird auf gesetzt, `true` um das Abonnement zu aktivieren, gesetzt auf, `false` um das Abonnement zu erstellen, aber nicht zu aktivieren.

Typ: Boolesch

Erforderlich: Nein

EventCategories. EventCategoryN

Eine Liste der Veranstaltungskategorien für eine`SourceType`, die Sie abonnieren möchten.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SourceIds. SourceIdN

Die Liste der IDs der Ereignisquellen, für die Ereignisse zurückgegeben werden. Wenn nicht angegeben, werden alle Quellen zur Antwort hinzugefügt. Eine ID muss mit einem Buchstaben beginnen und darf nur ASCII-Buchstaben, Ziffern und Bindestriche enthalten; sie darf nicht mit einem Bindestrich oder zwei aufeinander folgenden Bindestrichen enden.

Einschränkungen:

- Wenn `SourceIds` angegeben, `SourceType` muss auch angegeben werden.
- Wenn der Quelltyp eine Instanz ist, `DBInstanceIdentifier` muss a angegeben werden.
- Wenn der Quelltyp eine Sicherheitsgruppe ist, `DBSecurityGroupName` muss a angegeben werden.
- Wenn der Quelltyp eine Parametergruppe ist, `DBParameterGroupName` muss a angegeben werden.
- Wenn der Quelltyp ein Snapshot ist, `DBSnapshotIdentifier` muss a angegeben werden.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SourceType

Der Typ der Quelle, die die Ereignisse generiert. Wenn Sie beispielsweise über Ereignisse informiert werden möchten, die von einer Instanz generiert wurden, würden Sie diesen Parameter auf setzen`db-instance`. Wenn der Wert nicht angegeben ist, werden alle Ereignisse zurückgegeben.

Gültige Werte: `db-instance`, `db-cluster`, `db-parameter-group`, `db-security-group`, `db-cluster-snapshot`

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem Event-Abonnement zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

EventSubscription

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

EventSubscriptionQuotaExceeded

Sie haben die maximale Anzahl von Event-Abonnements erreicht.

HTTP Status Code: 400

SNSInvalidTopic

Amazon SNS hat geantwortet, dass es ein Problem mit dem angegebenen Thema gibt.

HTTP Status Code: 400

SNSNoAuthorization

Sie sind nicht berechtigt, zum SNS-Thema Amazon Resource Name (ARN) zu veröffentlichen.

HTTP Status Code: 400

SNSTopicArnNotFound

Das SNS-Thema Amazon Resource Name (ARN) existiert nicht.

HTTP Status Code: 404

SourceNotFound

Die angeforderte Quelle konnte nicht gefunden werden.

HTTP Status Code: 404

SubscriptionAlreadyExist

Der angegebene Abonnementname ist bereits vorhanden.

HTTP Status Code: 400

SubscriptionCategoryNotFound

Die angegebene Kategorie ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateGlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen globalen Amazon DocumentDB-Cluster, der sich über mehrere AWS-Regionen erstrecken kann. Der globale Cluster enthält einen primären Cluster mit Lese- und Schreibfunktion und bis zu fünf schreibgeschützte sekundäre Cluster. Global Clusters verwendet speicherbasierte schnelle Replikation über Regionen hinweg mit Latenzen von weniger als einer Sekunde. Dabei wird eine dedizierte Infrastruktur verwendet, ohne dass die Leistung Ihrer Workloads beeinträchtigt wird.

Sie können einen globalen Cluster erstellen, der zunächst leer ist, und ihm dann einen primären und einen sekundären hinzufügen. Oder Sie können während des Erstellungsvorgangs einen vorhandenen Cluster angeben, der dann zum primären Cluster des globalen Clusters wird.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die Cluster-ID des neuen globalen Clusters.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

DatabaseName

Der Name für Ihre Datenbank mit bis zu 64 alphanumerischen Zeichen. Wenn Sie keinen Namen angeben, erstellt Amazon DocumentDB keine Datenbank in dem globalen Cluster, den Sie erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Die Löschschutzeinstellung für den neuen globalen Cluster. Der globale Cluster kann nicht gelöscht werden, wenn der Löschschutz aktiviert ist.

Typ: Boolesch

Erforderlich: Nein

Engine

Der Name der Datenbank-Engine, die für diesen Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die Engine-Version des globalen Clusters.

Typ: Zeichenfolge

Erforderlich: Nein

SourceDBClusterIdentifier

Der Amazon-Ressourcenname (ARN), der als primärer Cluster des globalen Clusters verwendet werden soll. Dieser Parameter ist optional.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Die Einstellung für die Speicherverschlüsselung für den neuen globalen Cluster.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

GlobalClusterAlreadyExistsFault

`GlobalClusterIdentifier` ist bereits vorhanden. Wählen Sie eine neue globale Cluster-ID (eindeutiger Name), um einen neuen globalen Cluster zu erstellen.

HTTP Status Code: 400

GlobalClusterQuotaExceededFault

Die Anzahl der globalen Cluster für dieses Konto hat bereits den zulässigen Höchstwert erreicht.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteDBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht einen zuvor bereitgestellten Cluster. Wenn Sie einen Cluster löschen, werden alle automatisierten Backups für diesen Cluster gelöscht und können nicht wiederhergestellt werden. Manuelle DB-Cluster-Snapshots des angegebenen Clusters werden nicht gelöscht.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Cluster-ID für den zu löschenden Cluster. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit einem vorhandenen übereinstimmen`DBClusterIdentifier`.

Typ: Zeichenfolge

Erforderlich: Ja

FinalDBSnapshotIdentifier

Die Cluster-Snapshot-ID des neuen Cluster-Snapshots, der erstellt wurde, wenn auf gesetzt `SkipFinalSnapshot` ist`false`.

> **Note**
>
> Die Angabe dieses Parameters und das Setzen des `SkipFinalShapshot` Parameters `true` auf führen zu einem Fehler.

Einschränkungen:

- Muss aus 1 bis 255 Buchstaben, Zahlen oder Bindestrichen bestehen.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

SkipFinalSnapshot

Bestimmt, ob ein letzter Cluster-Snapshot erstellt wird, bevor der Cluster gelöscht wird. Wenn angegeben, `true` wird kein Cluster-Snapshot erstellt. Wenn angegeben, `false` wird ein Cluster-Snapshot erstellt, bevor der DB-Cluster gelöscht wird.

> **Note**
>
> Wenn `SkipFinalSnapshot` ja`false`, müssen Sie einen `FinalDBSnapshotIdentifier` Parameter angeben.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterSnapshotAlreadyExistsFault

Sie haben bereits einen Cluster-Snapshot mit der angegebenen Kennung.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteDBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht eine angegebene Cluster-Parametergruppe. Die zu löschende Cluster-Parametergruppe kann keinem Cluster zugeordnet werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der Cluster-Parametergruppe.

Einschränkungen:

- Muss der Name einer vorhandenen Cluster-Parametergruppe sein.
- Sie können eine Standard-Cluster-Parametergruppe nicht löschen.
- Kann keinem Cluster zugeordnet werden.

Typ: Zeichenfolge

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

InvalidDBParameterGroupState

Die Parametergruppe wird verwendet, oder sie befindet sich in einem Status, der nicht gültig ist. Wenn Sie versuchen, die Parametergruppe zu löschen, können Sie sie nicht löschen, wenn sich die Parametergruppe in diesem Status befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteDBClusterSnapshot

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht einen Cluster-Snapshot. Wenn der Snapshot gerade kopiert wird, wird der Kopiervorgang beendet.

> **Note**
>
> Der Cluster-Snapshot muss sich in dem `available` Zustand befinden, in dem er gelöscht werden kann.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterSnapshotIdentifier

Die ID des Cluster-Snapshots, der gelöscht werden soll.

Einschränkungen: Muss der Name eines vorhandenen Cluster-Snapshots im `available` Status sein.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterSnapshot

Detaillierte Informationen zu einem Cluster-Snapshot.

Typ: DBClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteDBInstance

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht eine zuvor bereitgestellte Instanz.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die Instanz-ID für die zu löschende Instanz. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit dem Namen einer vorhandenen Instanz übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instanz.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

DBSnapshotAlreadyExists

`DBSnapshotIdentifier`wird bereits von einem vorhandenen Snapshot verwendet.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

SnapshotQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Snapshots überschreiten.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteDBSubnetGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht eine Subnetzgruppe.

> **Note**
>
> Die angegebene Datenbanksubnetzgruppe muss nicht mit beliebigen DB-Instances verknüpft werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupName

Der Name der zu löschenden Datenbanksubnetzgruppe.

> **Note**
>
> Sie können die Standardsubnetzgruppe nicht löschen.

Einschränkungen:

Muss dem Namen einer vorhandenen `DBSubnetGroup` entsprechen. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

InvalidDBSubnetGroupStateFault

Die Subnetzgruppe kann nicht gelöscht werden, da sie verwendet wird.

HTTP Status Code: 400

InvalidDBSubnetStateFault

Das Subnetz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteEventSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht ein Abonnement für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SubscriptionName

Der Name des Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen, das Sie löschen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InvalidEventSubscriptionState

Möglicherweise ändert jemand anderes ein Abonnement. Warten Sie ein paar Sekunden und versuchen Sie es erneut.

HTTP Status Code: 400

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteGlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Löscht einen globalen Cluster. Der primäre und der sekundäre Cluster müssen bereits getrennt oder gelöscht sein, bevor versucht wird, einen globalen Cluster zu löschen.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die Cluster-ID des globalen Clusters, der gelöscht wird.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

GlobalClusterNotFoundFault

Der bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, solange sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeCertificates

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von Zertifikaten der Zertifizierungsstelle (CA) zurück, die Amazon DocumentDB dafür AWS-Konto bereitgestellt hat.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

CertificateIdentifier

Die vom Benutzer angegebene Zertifikats-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen für das angegebene Zertifikat zurückgegeben. Wenn dieser Parameter weggelassen wird, wird eine Liste mit bis zu `MaxRecords` Zertifikaten zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Beschränkungen

- Muss mit einem vorhandenen übereinstimmen`CertificateIdentifier`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen `DescribeCertificates`-Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehrere Datensätze vorhanden sind, als der Wert `MaxRecords` angibt, ist ein Paginierungstoken mit dem Namen einer Markierung in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen:

- Mindestbestellmenge: 20
- Maximum: 100

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

Zertifikate.Zertifikate.N

Eine Liste von Zertifikaten dafür. AWS-Konto

Typ: Array von Certificate-Objekten

Marker

Ein optionales Paginierungstoken, das bereitgestellt wird, wenn die Anzahl der abgerufenen Datensätze größer als `MaxRecords` ist. Wenn dieser Parameter angegeben ist, gibt die Markierung den nächsten Datensatz in der Liste an. Wenn Sie den Wert von `Marker` in den nächsten Aufruf aufnehmen, `DescribeCertificates` erhalten Sie die nächste Seite mit Zertifikaten.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

CertificateNotFound

`CertificateIdentifier`bezieht sich nicht auf ein vorhandenes Zertifikat.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBClusterParameterGroups

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von `DBClusterParameterGroup`-Beschreibungen zurück. Wenn ein `DBClusterParameterGroupName` Parameter angegeben ist, enthält die Liste nur die Beschreibung der angegebenen Cluster-Parametergruppe.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name einer bestimmten Cluster-Parametergruppe, für die Details zurückgegeben werden sollen.

Einschränkungen:

- Falls angegeben, muss er mit dem Namen einer vorhandenen Datei übereinstimmen`DBClusterParameterGroup`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

DBClusterParameterGroups. DBClusterParameterGroupN

Eine Liste von Cluster-Parametergruppen.

Typ: Array von DBClusterParameterGroup-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBClusterParameters

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt die detaillierte Parameterliste für eine bestimmte Cluster-Parametergruppe zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name einer bestimmten Cluster-Parametergruppe, für die Parameterdetails zurückgegeben werden sollen.

Einschränkungen:

- Falls angegeben, muss er mit dem Namen einer vorhandenen Datei übereinstimmen`DBClusterParameterGroup`.

Typ: Zeichenfolge

Erforderlich: Ja

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Source

Dieser Wert gibt an, dass nur die Parameter für eine bestimmte Quelle zurückgegeben werden sollen. Parameterquellen können `engine`, `service` oder `customer` sein.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Parameter.Parameter.N

Gibt eine Liste der Parameter für die Cluster-Parametergruppe zurück.

Typ: Array von Parameter-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBClusters

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen über bereitgestellte Amazon DocumentDB-Cluster zurück. Dieser API-Vorgang unterstützt die Paginierung. Für bestimmte Verwaltungsfunktionen wie Cluster- und Instance-Lebenszyklusmanagement nutzt Amazon DocumentDB Betriebstechnologie, die mit Amazon RDS und Amazon Neptune gemeinsam genutzt wird. Verwenden Sie den `filterName=engine,Values=docdb` Filterparameter, um nur Amazon DocumentDB-Cluster zurückzugeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die vom Benutzer angegebene Cluster-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen aus dem spezifischen Cluster zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Falls angegeben, muss er mit einem vorhandenen übereinstimmen`DBClusterIdentifier`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Ein Filter, der einen oder mehrere zu beschreibende Cluster angibt.

Unterstützte Filter:

- `db-cluster-id`- Akzeptiert Cluster-Identifikatoren und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur Informationen zu den Clustern, die durch diese ARNs identifiziert wurden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

DBClusters. DBClusterN

Eine Liste von Clustern.

Typ: Array von DBCluster-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBClusterSnapshotAttributes

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von Cluster-Snapshot-Attributnamen und -werten für einen manuellen DB-Cluster-Snapshot zurück.

Wenn Sie Snapshots mit anderen teilen AWS-Konten, werden das `restore` Attribut und eine Liste der IDs Personen `DescribeDBClusterSnapshotAttributes` zurückgegeben AWS-Konten , die berechtigt sind, den manuellen Cluster-Snapshot zu kopieren oder wiederherzustellen. Wenn `all` es in der Werteliste für das `restore` Attribut enthalten ist, ist der manuelle Cluster-Snapshot öffentlich und kann von allen AWS-Konten kopiert oder wiederhergestellt werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterSnapshotIdentifier

Der Bezeichner für den Cluster-Snapshot, für den die Attribute beschrieben werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterSnapshotAttributesResult

Detaillierte Informationen zu den Attributen, die einem Cluster-Snapshot zugeordnet sind.

Typ: DBClusterSnapshotAttributesResult Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBClusterSnapshots

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen über Cluster-Snapshots zurück. Dieser API-Vorgang unterstützt die Paginierung.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die ID des Clusters, für den die Liste der Cluster-Snapshots abgerufen werden soll. Dieser Parameter kann nicht mit dem `DBClusterSnapshotIdentifier` Parameter verwendet werden. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Falls angegeben, muss er mit dem Bezeichner eines vorhandenen identisch sein`DBCluster`.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterSnapshotIdentifier

Eine spezifische Cluster-Snapshot-ID, die beschrieben werden soll. Dieser Parameter kann nicht mit dem `DBClusterIdentifier` Parameter verwendet werden. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Falls angegeben, muss er mit dem Bezeichner eines vorhandenen identisch sein`DBClusterSnapshot`.
- Wenn diese Kennung für einen automatisierten Snapshot ist, muss auch der Parameter `SnapshotType` angegeben werden.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

IncludePublic

Ist auf eingestellt, `true` um manuelle Cluster-Snapshots einzubeziehen, die öffentlich sind und von jedem kopiert oder wiederhergestellt werden können, oder auf andere Weise. AWS-Konto`false` Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

IncludeShared

Legt fest, `true` dass auch gemeinsam genutzte manuelle Cluster-Snapshots von anderen Benutzern aufgenommen werden AWS-Konten , für AWS-Konto deren Kopieren oder Wiederherstellung die entsprechenden Rechte erteilt wurden, und aus anderen Gründen. `false` Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, wird ein Paginierungstoken (Markierung) in die Antwort aufgenommen, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

SnapshotType

Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:

- `automated`- Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Sie erstellt hat. AWS-Konto
- `manual`- Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihren erstellt haben. AWS-Konto
- `shared`- Gibt alle manuellen Cluster-Snapshots zurück, die für Sie freigegeben wurden. AWS-Konto
- `public`- Gibt alle Cluster-Snapshots zurück, die als öffentlich markiert wurden.

Wenn Sie keinen `SnapshotType` Wert angeben, werden sowohl automatisierte als auch manuelle Cluster-Snapshots zurückgegeben. Sie können gemeinsam genutzte Cluster-Snapshots in diese Ergebnisse einbeziehen, indem Sie den `IncludeShared` Parameter auf setzen. `true` Sie können öffentliche Cluster-Snapshots in diese Ergebnisse einbeziehen, indem Sie den `IncludePublic` Parameter auf setzen. `true`

Die Parameter `IncludePublic` und `IncludeShared` gelten nicht für `SnapshotType`-Werte von `manual` oder `automated`. Der Parameter `IncludePublic` gilt nicht, wenn `SnapshotType` auf `shared` festgelegt ist. Der Parameter `IncludeShared` gilt nicht, wenn `SnapshotType` auf `public` festgelegt ist.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

DBClusterSchnappschüsse. DBClusterSchnappschuss.N

Stellt eine Liste von Cluster-Snapshots bereit.

Typ: Array von DBClusterSnapshot-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBEngineVersions

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste der verfügbaren Engines zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBParameterGroupFamily

Der Name einer bestimmten Parametergruppenfamilie, für die Details zurückgegeben werden sollen.

Einschränkungen:

- Falls angegeben, muss er mit einem vorhandenen übereinstimmen`DBParameterGroupFamily`.

Typ: Zeichenfolge

Erforderlich: Nein

DefaultOnly

Gibt an, dass nur die standardmäßige Version der angegebenen Engine oder Kombination aus Engine und Hauptversion zurückgegeben wird.

Typ: Boolesch

Erforderlich: Nein

Engine

Die zurückzugebende Datenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die zurückzugebende Datenbank-Engine-Version.

Beispiel: `3.6.0`

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

ListSupportedCharacterSets

Wenn dieser Parameter angegeben ist und die angeforderte Engine den Parameter `CharacterSetName` für `CreateDBInstance` unterstützt, enthält die Antwort eine Liste der unterstützten Zeichensätze für jede Engine-Version.

Typ: Boolesch

Erforderlich: Nein

ListSupportedTimezones

Wenn dieser Parameter angegeben ist und die angeforderte Engine den Parameter `TimeZone` für `CreateDBInstance` unterstützt, enthält die Antwort eine Liste der unterstützten Zeitzonen für jede Engine-Version.

Typ: Boolesch

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

DBEngineVersionen. DBEngineVersion.N

Detaillierte Informationen zu einer oder mehreren Engine-Versionen.

Typ: Array von DBEngineVersion-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBInstances

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen über bereitgestellte Amazon DocumentDB DocumentDB-Instances zurück. Diese API unterstützt Paginierung.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die vom Benutzer angegebene Instance-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen von der jeweiligen Instanz zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Falls angegeben, muss er mit dem Bezeichner einer vorhandenen übereinstimmen`DBInstance`.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Ein Filter, der eine oder mehrere zu beschreibende Instanzen angibt.

Unterstützte Filter:

- `db-cluster-id`- Akzeptiert Cluster-Identifikatoren und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur die Informationen über die Instances, die den Clustern zugeordnet sind, die durch diese ARNs identifiziert werden.
- `db-instance-id`- Akzeptiert Instanzkennungen und Instanzen ARNs. Die Ergebnisliste enthält nur die Informationen über die Instanzen, die durch diese ARNs identifiziert werden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

DBInstances. DBInstanceN

Detaillierte Informationen zu einer oder mehreren Instanzen.

Typ: Array von DBInstance-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeDBSubnetGroups

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von `DBSubnetGroup`-Beschreibungen zurück. Wenn a angegeben `DBSubnetGroupName` ist, enthält die Liste nur die Beschreibungen der angegebenen`DBSubnetGroup`.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupName

Der Name der Subnetzgruppe, für die Details zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Nein

Filters.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

DBSubnetGruppen. DBSubnetGruppe.N

Detaillierte Informationen zu einer oder mehreren Subnetzgruppen.

Typ: Array von DBSubnetGroup-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeEngineDefaultClusterParameters

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt die Standard-Engine- und System-Parameterinformationen für die Cluster-Datenbank-Engine zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBParameterGroupFamily

Der Name der Cluster-Parametergruppenfamilie, für die die Engine-Parameterinformationen zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

EngineDefaults

Enthält das Ergebnis eines erfolgreichen Aufrufs der `DescribeEngineDefaultClusterParameters` Operation.

Typ: EngineDefaults Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python

- AWS SDK for Ruby V3

DescribeEventCategories

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Zeigt eine Liste der Kategorien für alle Ereignisquelltypen oder – falls angegeben – für einen angegebenen Quelltyp an.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

SourceType

Der Typ der Quelle, die die Ereignisse generiert.

Zulässige Werte: `db-instance`, `db-parameter-group`, `db-security-group`

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

EventCategoriesMapList. EventCategoriesMapN

Eine Liste von Karten mit Event-Kategorien.

Typ: Array von EventCategoriesMap-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeEvents

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt Ereignisse zurück, die sich auf Instances, Sicherheitsgruppen, Snapshots und DB-Parametergruppen der letzten 14 Tage beziehen. Sie können Ereignisse abrufen, die für eine bestimmte DB-Instance, Sicherheitsgruppe, einen Snapshot oder eine Parametergruppe spezifisch sind, indem Sie den Namen als Parameter angeben. Standardmäßig werden die Ereignisse der letzten Stunde zurückgegeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Duration

Die Anzahl der Minuten, in denen Ereignisse abgerufen werden sollen.

Standard: 60

Typ: Ganzzahl

Erforderlich: Nein

EndTime

Das Ende des Zeitintervalls, für das Ereignisse abgerufen werden sollen, angegeben im ISO 8601-Format.

Beispiel: 2009-07-08T18:00Z

Typ: Zeitstempel

Erforderlich: Nein

EventCategories. EventCategoryN

Eine Liste von Ereigniskategorien, die Benachrichtigungen für ein Abonnement für Ereignisbenachrichtigungen auslösen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

SourceIdentifier

ID der Ereignisquelle, für die Ereignisse zurückgegeben werden. Wenn nicht angegeben, werden alle Quellen zur Antwort hinzugefügt.

Einschränkungen:

- Wenn `SourceIdentifier` angegeben, `SourceType` muss es auch angegeben werden.
- Wenn der Quelltyp ist`DBInstance`, `DBInstanceIdentifier` muss a angegeben werden.
- Wenn der Quelltyp ist`DBSecurityGroup`, `DBSecurityGroupName` muss a angegeben werden.

- Wenn der Quelltyp ist`DBParameterGroup`, `DBParameterGroupName` muss a angegeben werden.
- Wenn der Quelltyp ist`DBSnapshot`, `DBSnapshotIdentifier` muss a angegeben werden.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

SourceType

Die Ereignisquelle zum Abrufen von Ereignissen. Wenn kein Wert angegeben ist, werden alle Ereignisse zurückgegeben.

Typ: Zeichenfolge

Zulässige Werte: `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Erforderlich: Nein

StartTime

Der Beginn des Zeitintervalls, für das Ereignisse abgerufen werden sollen, angegeben im ISO 8601-Format.

Beispiel: 2009-07-08T18:00Z

Typ: Zeitstempel

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

Events.Event.N

Detaillierte Informationen zu einer oder mehreren Veranstaltungen.

Typ: Array von Event-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeEventSubscriptions

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Listet alle Abonnementbeschreibungen für ein Kundenkonto auf. Die Beschreibung für ein Abonnement umfasst `SubscriptionNameSNSTopicARN`,`CustomerID`,`SourceType`,`SourceID`,`CreationTime`, und`Status`.

Wenn Sie eine angeben`SubscriptionName`, wird die Beschreibung für dieses Abonnement aufgeführt.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

SubscriptionName

Der Name des Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen, das Sie beschreiben möchten.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Service zurückgegeben.

EventSubscriptionsList. EventSubscriptionN

Eine Liste von Event-Abonnements.

Typ: Array von EventSubscription-Objekten

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeGlobalClusters

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt Informationen über globale Amazon DocumentDB-Cluster zurück. Diese API unterstützt Paginierung.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filters.Filter.N

Ein Filter, der einen oder mehrere globale DB-Cluster spezifiziert, die beschrieben werden sollen.

Unterstützte Filter: `db-cluster-id` akzeptiert Cluster-Identifikatoren und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur Informationen zu den Clustern, die durch diese ARNs identifiziert wurden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

GlobalClusterIdentifier

Die vom Benutzer angegebene Cluster-ID. Wenn dieser Parameter angegeben ist, werden nur Informationen aus dem spezifischen Cluster zurückgegeben. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen `DescribeGlobalClusters`-Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken, das als Markierung bezeichnet wird, in der Antwort enthalten, sodass Sie die verbleibenden Ergebnisse abrufen können.

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

GlobalClusters. GlobalClusterMemberN

Typ: Array von GlobalCluster-Objekten

Marker

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

GlobalClusterNotFoundFault

Das bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribeOrderableDBInstanceOptions

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste bestellbarer Instanzoptionen für die angegebene Engine zurück.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Engine

Der Name der Engine, für die Instanzoptionen abgerufen werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

DBInstanceClass

Der Filterwert der Instanzklasse. Geben Sie diesen Parameter an, um nur die verfügbaren Angebote anzuzeigen, die der angegebenen Instanzklasse entsprechen.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Der Filterwert der Engine-Version. Geben Sie diesen Parameter an, um nur die verfügbaren Angebote anzuzeigen, die der angegebenen Engine-Version entsprechen.

Typ: Zeichenfolge

Erforderlich: Nein

Filter.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

LicenseModel

Der Filterwert des Lizenzmodells. Geben Sie diesen Parameter an, um nur die verfügbaren Angebote anzuzeigen, die dem angegebenen Lizenzmodell entsprechen.

Typ: Zeichenfolge

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

Vpc

Der Wert des VPC-Filters (Virtual Private Cloud). Geben Sie diesen Parameter an, um nur die verfügbaren VPC- oder Nicht-VPC-Angebote anzuzeigen.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Bestellbare Optionen.Bestellbare DBInstance Option.N DBInstance

Die Optionen, die für eine bestimmte bestellbare Instanz verfügbar sind.

Typ: Array von OrderableDBInstanceOption-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DescribePendingMaintenanceActions

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt eine Liste von Ressourcen (z. B. Instanzen) zurück, für die mindestens eine Wartungsaktion aussteht.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

Filter.Filter.N

Ein Filter, der eine oder mehrere Ressourcen angibt, für die ausstehende Wartungsaktionen zurückgegeben werden sollen.

Unterstützte Filter:

- `db-cluster-id`- Akzeptiert Cluster-Identifikatoren und Cluster-Amazon-Ressourcennamen (ARNs). Die Ergebnisliste enthält nur ausstehende Wartungsaktionen für die Cluster, die durch diese ARNs identifiziert wurden.
- `db-instance-id`- Akzeptiert Instanzkennungen und Instanzen ARNs. Die Ergebnisliste enthält nur ausstehende Wartungsaktionen für die DB-Instances, die durch diese ARNs identifiziert wurden.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

MaxRecords

Die maximale Anzahl der in der Antwort zurückgegebenen Datensätze. Wenn mehr Datensätze als der angegebene `MaxRecords` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Standard: 100

Einschränkungen: Mindestwert 20, Höchstwert 100.

Typ: Ganzzahl

Erforderlich: Nein

ResourceIdentifier

Der ARN einer Ressource, für die ausstehende Wartungsaktionen zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Die folgenden Elemente werden vom Dienst zurückgegeben.

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

PendingMaintenanceActions. ResourcePendingMaintenanceActionsN

Die anzuwendenden Wartungsmaßnahmen.

Typ: Array von ResourcePendingMaintenanceActions-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

ResourceNotFoundFault

Die angegebene Ressourcen-ID wurde nicht gefunden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

FailoverDBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erzwingt einen Failover für einen Cluster.

Bei einem Failover für einen Cluster wird eine der Amazon DocumentDB DocumentDB-Repliken (schreibgeschützte Instances) im Cluster zur primären Instance (Cluster-Writer) heraufgestuft.

Wenn die primäre Instance ausfällt, wechselt Amazon DocumentDB automatisch zu einem Amazon DocumentDB DocumentDB-Replikat, falls eines existiert. Sie können ein Failover erzwingen, wenn Sie einen Ausfall einer primären Instance zum Testen simulieren möchten.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Eine Cluster-ID, für die ein Failover erzwungen werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Nein

TargetDBInstanceIdentifier

Der Name der Instance, die zur primären Instance hochgestuft werden soll.

Sie müssen die Instance-ID für ein Amazon DocumentDB DocumentDB-Replikat im Cluster angeben. Beispiel, `mydbcluster-replica1`.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin

- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

FailoverGlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Macht den angegebenen sekundären DB-Cluster zum primären DB-Cluster im globalen Cluster, wenn ein Failover eines globalen Clusters auftritt.

Verwenden Sie diesen Vorgang, um auf ein ungeplantes Ereignis zu reagieren, z. B. eine regionale Katastrophe in der Primärregion. Ein Failover kann zum Verlust von Schreibtransaktionsdaten führen, die vor dem Eintreten des Failover-Ereignisses nicht auf die gewählte Sekundärseite repliziert wurden. Der Wiederherstellungsprozess, der eine DB-Instance auf dem ausgewählten sekundären DB-Cluster zur primären Writer-DB-Instance hochstuft, garantiert jedoch, dass sich die Daten in einem transaktionskonsistenten Zustand befinden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die ID des globalen Amazon DocumentDB-Clusters, auf den dieser Vorgang angewendet werden soll. Die Kennung ist der eindeutige Schlüssel, den der Benutzer bei der Erstellung des Clusters zugewiesen hat. Mit anderen Worten, es ist der Name des globalen Clusters.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen globalen Clusters übereinstimmen.
- Mindestlänge 1. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

TargetDbClusterIdentifier

Die ID des sekundären Amazon DocumentDB-Clusters, den Sie zum primären für den globalen Cluster heraufstufen möchten. Verwenden Sie den Amazon Resource Name (ARN) für die Kennung, damit Amazon DocumentDB den Cluster in seiner AWS Region lokalisieren kann.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen sekundären Clusters übereinstimmen.
- Mindestlänge 1. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

AllowDataLoss

Gibt an, ob Datenverlust für diesen globalen Clustervorgang zulässig ist. Wenn Datenverlust zugelassen wird, wird ein globaler Failover-Vorgang ausgelöst.

Wenn Sie nichts angeben`AllowDataLoss`, wird für den globalen Clustervorgang standardmäßig ein Switchover verwendet.

Einschränkungen:

- Kann nicht zusammen mit dem Parameter angegeben werden. `Switchover`

Typ: Boolesch

Erforderlich: Nein

Switchover

Gibt an, ob dieser globale Datenbankcluster umgeschaltet werden soll.

Einschränkungen:

- Kann nicht zusammen mit dem `AllowDataLoss` Parameter angegeben werden.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

GlobalClusterNotFoundFault

Das bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, solange sich der Cluster in diesem Status befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ListTagsForResource

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Listet alle Tags auf einer Amazon DocumentDB DocumentDB-Ressource auf.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ResourceName

Die Amazon DocumentDB DocumentDB-Ressource mit Tags, die aufgelistet werden sollen. Dieser Wert ist ein Amazon-Ressourcenname (ARN).

Typ: Zeichenfolge

Erforderlich: Ja

Filters.Filter.N

Dieser Parameter wird derzeit nicht unterstützt.

Typ: Array von Filter-Objekten

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

TagList.Tag.N

Eine Liste mit einem oder mehreren Tags.

Typ: Array von Tag-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyDBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändert eine Einstellung für einen Amazon DocumentDB-Cluster. Sie können einen oder mehrere Datenbank-Konfigurationsparameter ändern, indem Sie diese Parameter und die neuen Werte in der Anforderung angeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Cluster-ID für den Cluster, der geändert wird. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

AllowMajorVersionUpgrade

Ein Wert, der angibt, ob Major-Versionsupgrades erlaubt sind.

Einschränkungen: Sie müssen größere Versionsupgrades zulassen, wenn Sie einen Wert für den `EngineVersion` Parameter angeben, der eine andere Hauptversion als die aktuelle Version des DB-Clusters ist.

Typ: Boolesch

Erforderlich: Nein

ApplyImmediately

Ein Wert, der angibt, ob die Änderungen in dieser Anforderung und alle ausstehenden Änderungen unabhängig von der `PreferredMaintenanceWindow` Einstellung für den Cluster so schnell wie möglich asynchron angewendet werden. Wenn dieser Parameter auf gesetzt ist`false`, werden Änderungen am Cluster im nächsten Wartungsfenster übernommen.

Der `ApplyImmediately` Parameter wirkt sich nur auf die `MasterUserPassword` Werte `NewDBClusterIdentifier` und aus. Wenn Sie diesen Parameterwert auf setzen`false`, werden die Änderungen an den `MasterUserPassword` Werten `NewDBClusterIdentifier` und im nächsten Wartungsfenster übernommen. Alle anderen Änderungen werden sofort übernommen, unabhängig von dem Wert des Parameters `ApplyImmediately`.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

BackupRetentionPeriod

Die Anzahl von Tagen, über die hinweg automatische Sicherungen aufbewahrt werden. Sie müssen einen Mindestwert von 1 angeben.

Standard: 1

Einschränkungen:

- Muss ein Wert zwischen 1 und 35 sein.

Typ: Ganzzahl

Erforderlich: Nein

CloudwatchLogsExportConfiguration

Die Konfigurationseinstellung für die Protokolltypen, die für den Export nach Amazon CloudWatch Logs für eine bestimmte Instance oder einen bestimmten Cluster aktiviert werden sollen. Die `DisableLogTypes` Arrays `EnableLogTypes` und bestimmen, welche Protokolle in Logs exportiert (oder nicht exportiert) werden CloudWatch .

Typ: CloudwatchLogsExportConfiguration Objekt

Erforderlich: Nein

DBClusterParameterGroupName

Der Name der Cluster-Parametergruppe, die für den Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn `DeletionProtection` aktiviert, kann der Cluster nur gelöscht werden, wenn er geändert und `DeletionProtection` deaktiviert wurde. `DeletionProtection`schützt Cluster davor, versehentlich gelöscht zu werden.

Typ: Boolesch

Erforderlich: Nein

EngineVersion

Die Versionsnummer der Datenbank-Engine, auf die ein Upgrade durchgeführt werden soll. Das Ändern dieses Parameters führt zu einem Nutzungsausfall. Die Änderung wird im nächsten Wartungsfenster übernommen, es sei denn, `ApplyImmediately` ist aktiviert.

Verwenden Sie den folgenden Befehl, um alle verfügbaren Engine-Versionen für Amazon DocumentDB aufzulisten:

```
aws docdb describe-db-engine-versions --engine docdb --query
"DBEngineVersions[].EngineVersion"
```

Typ: Zeichenfolge

Erforderlich: Nein

ManageMasterUserPassword

Gibt an, ob das Masterbenutzerpasswort mit Amazon Web Services Secrets Manager verwaltet werden soll. Wenn der Cluster das Masterbenutzerkennwort nicht mit Amazon Web Services Secrets Manager verwaltet, können Sie diese Verwaltung aktivieren. In diesem Fall können Sie nichts angeben`MasterUserPassword`. Wenn der Cluster das Masterbenutzer-Passwort bereits mit Amazon Web Services Secrets Manager verwaltet und Sie angeben, dass das Master-Benutzerpasswort nicht mit Amazon Web Services Secrets Manager verwaltet wird, müssen Sie angeben`MasterUserPassword`. In diesem Fall löscht Amazon DocumentDB das Geheimnis und verwendet das neue Passwort für den Masterbenutzer, das von angegeben wurde. `MasterUserPassword`

Typ: Boolesch

Erforderlich: Nein

MasterUserPassword

Das Passwort für den Masterbenutzer der Datenbank. Dieses Passwort kann alle druckbaren ASCII-Zeichen, außer Schrägstrich (/), doppeltes Anführungszeichen (") oder das "At"-Zeichen (@), enthalten.

Einschränkungen: Muss 8 bis 100 Zeichen enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserSecretKmsKeyId

Die Amazon Web Services KMS-Schlüssel-ID zur Verschlüsselung eines Geheimnisses, das automatisch im Amazon Web Services Secrets Manager generiert und verwaltet wird.

Diese Einstellung ist nur gültig, wenn die beiden folgenden Bedingungen erfüllt sind:

- Der Cluster verwaltet das Masterbenutzer-Passwort in Amazon Web Services Secrets Manager nicht. Wenn der Cluster das Masterbenutzerkennwort bereits in Amazon Web Services Secrets Manager verwaltet, können Sie den KMS-Schlüssel, der zur Verschlüsselung des Geheimnisses verwendet wird, nicht ändern.
- Sie aktivieren `ManageMasterUserPassword` die Verwaltung des Masterbenutzer-Passworts in Amazon Web Services Secrets Manager. Wenn Sie das Gerät einschalten `ManageMasterUserPassword` und nichts angeben`MasterUserSecretKmsKeyId`, wird der `aws/secretsmanager` KMS-Schlüssel zur Verschlüsselung des Geheimnisses verwendet. Wenn sich das Geheimnis in einem anderen Amazon Web Services Services-Konto befindet, können Sie den `aws/secretsmanager` KMS-Schlüssel nicht verwenden, um das Geheimnis zu verschlüsseln, und Sie müssen einen vom Kunden verwalteten KMS-Schlüssel verwenden.

Die Amazon Web Services KMS-Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel. Um einen KMS-Schlüssel in einem anderen Amazon Web Services Services-Konto zu verwenden, geben Sie den Schlüssel-ARN oder den Alias-ARN an.

Es gibt einen Standard-KMS-Schlüssel für Ihr Amazon Web Services Services-Konto. Ihr Amazon Web Services Services-Konto hat für jede Amazon Web Services Services-Region einen anderen Standard-KMS-Schlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

NewDBClusterIdentifier

Die neue Cluster-ID für den Cluster beim Umbenennen eines Clusters. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster2`

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Nummer des Ports, an dem der Cluster Verbindungen akzeptiert.

Einschränkungen: Muss ein Wert von `1150` bis sein`65535`.

Standard: Derselbe Port wie der ursprüngliche Cluster.

Typ: Ganzzahl

Erforderlich: Nein

PreferredBackupWindow

Der tägliche Zeitraum, in dem automatische Sicherungen erstellt werden, wenn diese mit dem Parameter `BackupRetentionPeriod` aktiviert sind.

Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils AWS-Region 8 Stunden ausgewählt wird.

Einschränkungen:

- Muss im Format `hh24:mi-hh24:mi` angegeben werden.
- Muss in Universal Coordinated Time (UTC) angegeben werden.
- Darf nicht mit dem bevorzugten Wartungsfenster in Konflikt treten.
- Muss mindestens 30 Minuten betragen.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Die Standardeinstellung ist ein 30-minütiges Fenster, das nach dem Zufallsprinzip aus einem 8-Stunden-Zeitblock ausgewählt wird und an einem zufälligen Wochentag stattfindet. AWS-Region

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

RotateMasterUserPassword

Gibt an, ob das von Amazon Web Services Secrets Manager verwaltete Geheimnis für das Masterbenutzerkennwort rotiert werden soll.

Diese Einstellung ist nur gültig, wenn das Masterbenutzerkennwort von Amazon DocumentDB im Amazon Web Services Secrets Manager für den Cluster verwaltet wird. Der geheime Wert enthält das aktualisierte Passwort.

Einschränkung: Sie müssen die Änderung sofort anwenden, wenn Sie das Masterbenutzerkennwort ändern.

Typ: Boolesch

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Gültige Werte für den Speichertyp - `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

VpcSecurityGroupIds. VpcSecurityGroupIdN

Eine Liste der Virtual Private Cloud (VPC) -Sicherheitsgruppen, zu denen der Cluster gehören wird.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterParameterGroupNotFound

`DBClusterParameterGroupName`bezieht sich nicht auf eine bestehende Cluster-Parametergruppe.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

InvalidDBSecurityGroupState

Der Status der Sicherheitsgruppe lässt kein Löschen zu.

HTTP Status Code: 400

InvalidDBSubnetGroupStateFault

Die Subnetzgruppe kann nicht gelöscht werden, da sie verwendet wird.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nach ihrer Erstellung nicht alle Availability Zones ab, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyDBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändert die Parameter einer Cluster-Parametergruppe. Wenn Sie mehr als einen Parameter ändern möchten, übergeben Sie eine Liste mit den folgenden Optionen: `ParameterName`, `ParameterValue` und `ApplyMethod`. Pro Anforderung können maximal 20 Parameter geändert werden.

> **Note**
>
> Änderungen an dynamischen Parametern werden sofort angewendet. Änderungen an statischen Parametern erfordern einen Neustart oder ein Wartungsfenster, bevor die Änderung wirksam wird.

> **Important**
>
> Nachdem Sie eine Cluster-Parametergruppe erstellt haben, sollten Sie mindestens fünf Minuten warten, bevor Sie das erste Cluster erstellen, das diese Cluster-Parametergruppe als Standardparametergruppe verwendet. Dadurch kann Amazon DocumentDB die Erstellungsaktion vollständig abschließen, bevor die Parametergruppe als Standard für einen neuen Cluster verwendet wird. Dieser Schritt ist besonders wichtig für Parameter, die beim Erstellen der Standarddatenbank für einen Cluster von kritischer Bedeutung sind, z. B. den Zeichensatz für die Standarddatenbank, der durch den Parameter `character_set_database` definiert wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der zu ändernden Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

Parameters.Parameter.N

Eine Liste der Parameter in der Cluster-Parametergruppe, die geändert werden sollen.

Typ: Array von Parameter-Objekten

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterParameterGroupName

Der Name einer Cluster-Parametergruppe.

Einschränkungen:

- Muss aus 1 bis 255 Buchstaben oder Zahlen bestehen.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

> **Note**
>
> Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

InvalidDBParameterGroupState

Die Parametergruppe wird verwendet, oder sie befindet sich in einem Status, der nicht gültig ist. Wenn Sie versuchen, die Parametergruppe zu löschen, können Sie sie nicht löschen, wenn sich die Parametergruppe in diesem Status befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyDBClusterSnapshotAttribute

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Fügt einem manuellen Cluster-Snapshot ein Attribut und Werte hinzu oder entfernt ein Attribut und Werte daraus.

Um einen manuellen Cluster-Snapshot gemeinsam mit anderen zu verwenden AWS-Konten, geben Sie das `restore` als an und verwenden Sie den `ValuesToAdd` Parameter`AttributeName`, um eine Liste IDs der Personen hinzuzufügen AWS-Konten , die berechtigt sind, den manuellen Cluster-Snapshot wiederherzustellen. Verwenden Sie den Wert`all`, um den manuellen Cluster-Snapshot öffentlich zu machen, was bedeutet, dass er von allen kopiert oder wiederhergestellt werden kann AWS-Konten. Fügen Sie nicht den `all` Wert für manuelle Cluster-Snapshots hinzu, die private Informationen enthalten, die nicht für alle AWS-Konten verfügbar sein sollen. Wenn ein manueller Cluster-Snapshot verschlüsselt ist, kann er gemeinsam genutzt werden, jedoch nur, indem eine Liste der AWS-Konto IDs für den `ValuesToAdd` Parameter autorisierten Personen angegeben wird. Sie können in diesem Fall `all` nicht als Wert für diesen Parameter verwenden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

AttributeName

Der Name des Cluster-Snapshot-Attributs, das geändert werden soll.

Um die Autorisierung anderer AWS-Konten zum Kopieren oder Wiederherstellen eines manuellen Cluster-Snapshots zu verwalten, legen Sie diesen Wert auf fest`restore`.

Typ: Zeichenfolge

Erforderlich: Ja

DBClusterSnapshotIdentifier

Der Bezeichner für den Cluster-Snapshot, für den die Attribute geändert werden sollen.

Typ: Zeichenfolge

Erforderlich: Ja

ValuesToAdd. AttributeValueN

Eine Liste von Cluster-Snapshot-Attributen, die zu dem von `AttributeName` angegebenen Attribut hinzugefügt werden sollen.

Um andere Benutzer AWS-Konten zum Kopieren oder Wiederherstellen eines manuellen Cluster-Snapshots zu autorisieren, legen Sie fest, dass diese Liste einen oder mehrere AWS-Konto IDs enthält. Um den manuellen Cluster-Snapshot durch einen beliebigen Benutzer wiederherstellbar zu machen AWS-Konto, setzen Sie ihn auf. `all` Fügen Sie keinen `all` Wert für manuelle Cluster-Snapshots hinzu, die private Informationen enthalten, die nicht für alle verfügbar sein sollen. AWS-Konten

Typ: Zeichenfolgen-Array

Erforderlich: Nein

ValuesToRemove. AttributeValueN

Eine Liste von Cluster-Snapshot-Attributen, die aus dem von `AttributeName` angegebenen Attribut entfernt werden sollen.

Um anderen die Autorisierung AWS-Konten zum Kopieren oder Wiederherstellen eines manuellen Cluster-Snapshots zu entziehen, legen Sie fest, dass diese Liste einen oder mehrere AWS-Konto Identifikatoren enthält. Um anderen Benutzern die Autorisierung AWS-Konto zum Kopieren oder Wiederherstellen des Cluster-Snapshots zu entziehen, setzen Sie sie auf`all`. Wenn Sie angeben`all`, kann ein AWS-Konto Benutzer, dessen Konto-ID dem `restore` Attribut explizit hinzugefügt wurde, trotzdem einen manuellen Cluster-Snapshot kopieren oder wiederherstellen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterSnapshotAttributesResult

Detaillierte Informationen zu den Attributen, die einem Cluster-Snapshot zugeordnet sind.

Typ: DBClusterSnapshotAttributesResult Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

SharedSnapshotQuotaExceeded

Sie haben die maximale Anzahl an Konten überschritten, für die Sie einen manuellen DB-Snapshot freigeben können.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyDBInstance

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändert die Einstellungen für eine Instanz. Sie können einen oder mehrere Datenbank-Konfigurationsparameter ändern, indem Sie diese Parameter und die neuen Werte in der Anforderung angeben.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die Instance-ID. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBInstance` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

ApplyImmediately

Gibt an, ob die Änderungen in dieser Anforderung und alle ausstehenden Änderungen unabhängig von der `PreferredMaintenanceWindow` Einstellung für die Instanz so schnell wie möglich asynchron angewendet werden.

Wenn dieser Parameter auf gesetzt ist`false`, werden Änderungen an der Instanz im nächsten Wartungsfenster übernommen. Einige Parameteränderungen können zu einem Ausfall führen und werden beim nächsten Neustart übernommen.

Standard: `false`

Typ: Boolesch

Erforderlich: Nein

AutoMinorVersionUpgrade

Dieser Parameter gilt nicht für Amazon DocumentDB. Amazon DocumentDB führt unabhängig vom eingestellten Wert keine kleineren Versions-Upgrades durch.

Typ: Boolesch

Erforderlich: Nein

CACertificateIdentifier

Gibt das Zertifikat an, das mit der Instance verknüpft werden muss.

Typ: Zeichenfolge

Erforderlich: Nein

CertificateRotationRestart

Gibt an, ob die DB-Instance neu gestartet wird, wenn Sie Ihr SSL/TLS-Zertifikat rotieren.

Standardmäßig wird die DB-Instance neu gestartet, wenn Sie Ihr SSL/TLS-Zertifikat rotieren. Das Zertifikat wird erst aktualisiert, wenn die DB-Instance neu gestartet wird.

> **Important**
>
> Legen Sie diesen Parameter nur fest, wenn Sie SSL/TLS nicht verwenden, um eine Verbindung zur DB-Instance herzustellen.

Wenn Sie SSL/TLS verwenden, um eine Verbindung zur DB-Instance herzustellen, finden Sie weitere Informationen unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten bei der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Typ: Boolesch

Erforderlich: Nein

CopyTagsToSnapshot

Ein Wert, der angibt, ob alle Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden sollen. Standardmäßig werden Tags nicht kopiert.

Typ: Boolesch

Erforderlich: Nein

DBInstanceClass

Die neue Rechen- und Speicherkapazität der Instance; zum Beispiel`db.r5.large`. Nicht alle Instanzklassen sind in allen verfügbar AWS-Regionen.

Wenn Sie die Instanzklasse ändern, kommt es während der Änderung zu einem Ausfall. Die Änderung wird während des nächsten Wartungsfensters angewendet, es sei denn, `ApplyImmediately` wird für diese Anforderung als `true` angegeben.

Standard: Verwendet die vorhandene Einstellung.

Typ: Zeichenfolge

Erforderlich: Nein

EnablePerformanceInsights

Ein Wert, der angibt, ob Performance Insights für die DB-Instance aktiviert werden sollen. Weitere Informationen finden Sie unter Verwenden von Amazon Performance Insights.

Typ: Boolesch

Erforderlich: Nein

NewDBInstanceIdentifier

Die neue Instanz-ID für die Instanz beim Umbenennen einer Instanz. Wenn Sie die Instanz-ID ändern, erfolgt sofort ein Instanzneustart, wenn Sie dies `Apply Immediately` auf `true` einstellen. Er erfolgt während des nächsten Wartungsfensters, wenn Sie `Apply Immediately` auf eingestellt haben`false`. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `mydbinstance`

Typ: Zeichenfolge

Erforderlich: Nein

PerformanceInsightsKMSKeyId

Die AWS KMS Schlüssel-ID für die Verschlüsselung von Performance Insights Insights-Daten.

Die AWS KMS Schlüssel-ID ist der Schlüssel-ARN, die Schlüssel-ID, der Alias-ARN oder der Aliasname für den KMS-Schlüssel.

Wenn Sie keinen Wert für PerformanceInsights KMSKey Id angeben, verwendet Amazon DocumentDB Ihren Standard-KMS-Schlüssel. Es gibt einen Standard-KMS-Schlüssel für Ihr Amazon Web Services Services-Konto. Ihr Amazon Web Services Services-Konto hat für jede Amazon Web Services Services-Region einen anderen Standard-KMS-Schlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Der wöchentliche Zeitraum (in UTC), in dem Systemwartungen durchgeführt werden können, die möglicherweise zu einem Nutzungsausfall führen. Eine Änderung dieses Parameters führt nicht zu einem Ausfall, außer in der folgenden Situation, und die Änderung wird so schnell wie möglich asynchron angewendet. Wenn ausstehende Aktionen vorhanden sind, die einen Neustart verursachen, und das Wartungsfenster so geändert wird, dass es die aktuelle Uhrzeit enthält, führt eine Änderung dieses Parameters zu einem Neustart der Instanz. Wenn Sie dieses Fenster auf die aktuelle Uhrzeit verschieben, müssen zwischen der aktuellen Uhrzeit und dem Ende des Fensters mindestens 30 Minuten liegen, um sicherzustellen, dass ausstehende Änderungen übernommen werden.

Standard: Verwendet die vorhandene Einstellung.

Format: `ddd:hh24:mi-ddd:hh24:mi`

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: Muss mindestens 30 Minuten dauern.

Typ: Zeichenfolge

Erforderlich: Nein

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance heraufgestuft wird.

Standard: 1

Gültige Werte: 0-15

Typ: Ganzzahl

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instanz.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AuthorizationNotFound

Die angegebene CIDR-IP oder EC2 Amazon-Sicherheitsgruppe ist für die angegebene Sicherheitsgruppe nicht autorisiert.

Amazon DocumentDB ist möglicherweise auch nicht autorisiert, die erforderlichen Aktionen in Ihrem Namen mithilfe von IAM durchzuführen.

HTTP Status Code: 404

CertificateNotFound

`CertificateIdentifier`bezieht sich nicht auf ein vorhandenes Zertifikat.

HTTP Status Code: 404

DBInstanceAlreadyExists

Sie haben bereits eine Instanz mit der angegebenen ID.

HTTP Status Code: 400

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

DBSecurityGroupNotFound

`DBSecurityGroupName`bezieht sich nicht auf eine bestehende Sicherheitsgruppe.

HTTP Status Code: 404

DBUpgradeDependencyFailure

Das Upgrade ist fehlgeschlagen, weil eine Ressource, von der der abhängt, nicht geändert werden kann.

HTTP Status Code: 400

InsufficientDBInstanceCapacity

Die angegebene Instanzklasse ist in der angegebenen Availability Zone nicht verfügbar.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

InvalidDBSecurityGroupState

Der Status der Sicherheitsgruppe lässt kein Löschen zu.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nach ihrer Erstellung nicht alle Availability Zones ab, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instances verfügbar ist.

HTTP Status Code: 400

StorageTypeNotSupported

Der angegebene Speicher `StorageType` kann der DB-Instance nicht zugeordnet werden.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyDBSubnetGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändert eine bestehende Subnetzgruppe. Subnetzgruppen müssen mindestens ein Subnetz in mindestens zwei Availability Zones in der enthalten. AWS-Region

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBSubnetGroupName

Der Name für die Subnetzgruppe. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert. Sie können die Standardsubnetzgruppe nicht ändern.

Einschränkungen: Der Wert muss mit dem Namen einer vorhandenen `DBSubnetGroup` übereinstimmen. Der Name darf nicht default sein.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Ja

SubnetIds. SubnetIdentifierN

Das EC2 Amazon-Subnetz IDs für die Subnetzgruppe.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

DBSubnetGroupDescription

Die Beschreibung für die Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

DBSubnetGroup

Detaillierte Informationen zu einer Subnetzgruppe.

Typ: DBSubnetGroup Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBSubnetGroupDoesNotCoverEnoughAZs

Subnetze in der Subnetzgruppe sollten mindestens zwei Availability Zones abdecken, es sei denn, es gibt nur eine Availability Zone.

HTTP Status Code: 400

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

DBSubnetQuotaExceededFault

Die Anfrage würde dazu führen, dass Sie die zulässige Anzahl von Subnetzen in einer Subnetzgruppe überschreiten.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

SubnetAlreadyInUse

Das Subnetz wird bereits in der Availability Zone verwendet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyEventSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändert ein vorhandenes Abonnement für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SubscriptionName

Der Name des Abonnements für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Erforderlich: Ja

Enabled

Ein boolescher Wert; wird auf gesetzt, um das Abonnement `true` zu aktivieren.

Typ: Boolesch

Erforderlich: Nein

EventCategories. EventCategoryN

Eine Liste der Veranstaltungskategorien für eine`SourceType`, die Sie abonnieren möchten.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SnsTopicArn

Der Amazon-Ressourcenname (ARN) des SNS-Themas, das für die Ereignisbenachrichtigung erstellt wurde. Der ARN wird von Amazon SNS erstellt, wenn Sie ein Thema erstellen und es abonnieren.

Typ: Zeichenfolge

Erforderlich: Nein

SourceType

Der Typ der Quelle, die die Ereignisse generiert. Wenn Sie beispielsweise über Ereignisse informiert werden möchten, die von einer Instanz generiert wurden, setzen Sie diesen Parameter auf`db-instance`. Wenn der Wert nicht angegeben ist, werden alle Ereignisse zurückgegeben.

Zulässige Werte: `db-instance`, `db-parameter-group`, `db-security-group`

Typ: Zeichenfolge

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

EventSubscription

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

EventSubscriptionQuotaExceeded

Sie haben die maximale Anzahl von Event-Abonnements erreicht.

HTTP Status Code: 400

SNSInvalidTopic

Amazon SNS hat geantwortet, dass es ein Problem mit dem angegebenen Thema gibt.

HTTP Status Code: 400

SNSNoAuthorization

Sie sind nicht berechtigt, zum SNS-Thema Amazon Resource Name (ARN) zu veröffentlichen.

HTTP Status Code: 400

SNSTopicArnNotFound

Das SNS-Thema Amazon Resource Name (ARN) existiert nicht.

HTTP Status Code: 404

SubscriptionCategoryNotFound

Die angegebene Kategorie ist nicht vorhanden.

HTTP Status Code: 404

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ModifyGlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändern Sie eine Einstellung für einen globalen Amazon DocumentDB-Cluster. Sie können einen oder mehrere Konfigurationsparameter (zum Beispiel: Löschschutz) oder die globale Cluster-ID ändern, indem Sie diese Parameter und die neuen Werte in der Anfrage angeben.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Der Bezeichner für den globalen Cluster, der geändert wird. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der ID eines vorhandenen globalen Clusters übereinstimmen.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

DeletionProtection

Gibt an, ob für den globalen Cluster der Löschschutz aktiviert ist. Der globale Cluster kann nicht gelöscht werden, wenn der Löschschutz aktiviert ist.

Typ: Boolesch

Erforderlich: Nein

NewGlobalClusterIdentifier

Der neue Bezeichner für einen globalen Cluster, wenn Sie einen globalen Cluster ändern. Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.

  Das erste Zeichen muss ein Buchstabe sein.

  Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten

Beispiel: `my-cluster2`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

GlobalClusterNotFoundFault

Der bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, solange sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RebootDBInstance

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Möglicherweise müssen Sie Ihre Instance neu starten, normalerweise aus Wartungsgründen. Wenn Sie beispielsweise bestimmte Änderungen vornehmen oder wenn Sie die Cluster-Parametergruppe ändern, die der Instance zugeordnet ist, müssen Sie die Instance neu starten, damit die Änderungen wirksam werden.

Durch den Neustart einer Instance wird der Datenbank-Engine-Service neu gestartet. Der Neustart einer Instance führt zu einem kurzzeitigen Ausfall, bei dem der Instance-Status auf Neustart gesetzt wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBInstanceIdentifier

Die Instance-ID. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBInstance` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

ForceFailover

Wenn`true`, wird der Neustart über ein Multi-AZ-Failover durchgeführt.

Einschränkung: Sie können nicht angeben, `true` ob die Instance nicht für Multi-AZ konfiguriert ist.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBInstance

Detaillierte Informationen zu einer Instanz.

Typ: DBInstance Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RemoveFromGlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Trennt einen sekundären Amazon DocumentDB-Cluster von einem globalen Cluster. Der Cluster wird zu einem eigenständigen Cluster mit Lese- und Schreibfunktion, anstatt schreibgeschützt zu sein und Daten von einem Primärcluster in einer anderen Region zu empfangen.

> **Note**
>
> Diese Aktion gilt nur für Amazon DocumentDB-Cluster.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DbClusterIdentifier

Der Amazon-Ressourcenname (ARN), der den Cluster identifiziert, der vom globalen Amazon DocumentDB-Cluster getrennt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

GlobalClusterIdentifier

Die Cluster-ID, die vom globalen Amazon DocumentDB-Cluster getrennt werden soll.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

GlobalClusterNotFoundFault

Das bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, solange sich der Cluster in diesem Zustand befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3

- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RemoveSourceIdentifierFromSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Entfernt eine Quell-ID aus einem bestehenden Abonnement für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

SourceIdentifier

Die Quell-ID, die aus dem Abonnement entfernt werden soll, z. B. die Instance-ID für eine Instance oder der Name einer Sicherheitsgruppe.

Typ: Zeichenfolge

Erforderlich: Ja

SubscriptionName

Der Name des Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen, aus dem Sie eine Quell-ID entfernen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Service zurückgegeben.

EventSubscription

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Typ: EventSubscription Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

SourceNotFound

Die angeforderte Quelle konnte nicht gefunden werden.

HTTP Status Code: 404

SubscriptionNotFound

Der Abonnementname ist nicht vorhanden.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RemoveTagsFromResource

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Entfernt Metadaten-Tags aus einer Amazon DocumentDB DocumentDB-Ressource.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

ResourceName

Die Amazon DocumentDB DocumentDB-Ressource, aus der die Tags entfernt wurden. Dieser Wert ist ein Amazon-Ressourcenname (ARN).

Typ: Zeichenfolge

Erforderlich: Ja

TagKeys.Mitglied.N

Der Tag-Schlüssel (Name) des zu entfernenden Tags.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBInstanceNotFound

`DBInstanceIdentifier`bezieht sich nicht auf eine bestehende Instanz.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ResetDBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ändert die Parameter einer Cluster-Parametergruppe auf den Standardwert. Um bestimmte Parameter zurückzusetzen, reichen Sie eine Liste mit den folgenden Angaben ein: `ParameterName` und`ApplyMethod`. Um die gesamte Cluster-Parametergruppe zurückzusetzen, geben Sie die `ResetAllParameters` Parameter `DBClusterParameterGroupName` und an.

Wenn Sie die gesamte Gruppe zurücksetzen, werden die dynamischen Parameter sofort aktualisiert und die statischen Parameter werden so eingestellt, `pending-reboot` dass sie beim nächsten Neustart der DB-Instance wirksam werden.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterParameterGroupName

Der Name der zurückzusetzenden Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Ja

Parameters.Parameter.N

Eine Liste von Parameternamen in der Cluster-Parametergruppe, die auf die Standardwerte zurückgesetzt werden sollen. Sie können den Parameter nicht verwenden, wenn der Parameter `ResetAllParameters` auf `true` festgelegt ist.

Typ: Array von Parameter-Objekten

Erforderlich: Nein

ResetAllParameters

Ein Wert, auf den festgelegt ist`true`, um alle Parameter in der Cluster-Parametergruppe auf ihre Standardwerte zurückzusetzen und `false` andernfalls. Diesen Parameter können Sie nicht verwenden, wenn es eine Liste von Parameternamen für `Parameters` gibt.

Typ: Boolesch

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBClusterParameterGroupName

Der Name einer Cluster-Parametergruppe.

Einschränkungen:

- Muss aus 1 bis 255 Buchstaben oder Zahlen bestehen.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

> **Note**
>
> Dieser Wert wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBParameterGroupNotFound

`DBParameterGroupName`bezieht sich nicht auf eine bestehende Parametergruppe.

HTTP Status Code: 404

InvalidDBParameterGroupState

Die Parametergruppe wird verwendet, oder sie befindet sich in einem Status, der nicht gültig ist. Wenn Sie versuchen, die Parametergruppe zu löschen, können Sie sie nicht löschen, wenn sich die Parametergruppe in diesem Status befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RestoreDBClusterFromSnapshot

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Erstellt einen neuen Cluster aus einem Snapshot oder Cluster-Snapshot.

Wenn ein Snapshot angegeben ist, wird der Zielcluster aus dem Quell-DB-Snapshot mit einer Standardkonfiguration und einer Standardsicherheitsgruppe erstellt.

Wenn ein Cluster-Snapshot angegeben ist, wird der Zielcluster aus dem Quellclusterwiederherstellungspunkt mit derselben Konfiguration wie der ursprüngliche Quell-DB-Cluster erstellt, mit der Ausnahme, dass der neue Cluster mit der Standardsicherheitsgruppe erstellt wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Der Name des Clusters, der aus dem Snapshot oder Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-snapshot-id`

Typ: Zeichenfolge

Erforderlich: Ja

Engine

Die Datenbank-Engine, die für den neuen Cluster verwendet werden soll.

Standard: Entspricht der Quelle.

Einschränkung: Muss mit der Engine der Quelle kompatibel sein.

Typ: Zeichenfolge

Erforderlich: Ja

SnapshotIdentifier

Die Kennung für den Snapshot oder den Cluster-Snapshot, der zur Wiederherstellung verwendet werden soll.

Sie können entweder den Namen oder den Amazon-Ressourcennamen (ARN) verwenden, um einen Cluster-Snapshot festzulegen. Sie können jedoch auch nur den ARN verwenden, um einen Snapshot festzulegen.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen Snapshots übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

AvailabilityZones. AvailabilityZoneN

Stellt die Liste der Amazon EC2 Availability Zones bereit, in denen Instances im wiederhergestellten DB-Cluster erstellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

DBClusterParameterGroupName

Der Name der DB-Cluster-Parametergruppe, die mit diesem DB-Cluster verknüpft werden soll.

Typ: Zeichenfolge. Erforderlich: Nein.

Wenn dieses Argument weggelassen wird, wird die standardmäßige DB-Cluster-Parametergruppe verwendet. Falls angegeben, muss sie mit dem Namen einer vorhandenen Standard-DB-Cluster-Parametergruppe übereinstimmen. Die Zeichenfolge muss aus 1 bis 255 Buchstaben, Zahlen oder Bindestrichen bestehen. Das erste Zeichen muss ein Buchstabe sein und darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupName

Der Name der Subnetzgruppe, die für den neuen Cluster verwendet werden soll.

Einschränkungen: Falls angegeben, muss der Name mit dem Namen einer vorhandenen `DBSubnetGroup` Datei übereinstimmen.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn `DeletionProtection` aktiviert, kann der Cluster nur gelöscht werden, wenn er geändert und `DeletionProtection` deaktiviert wurde. `DeletionProtection`schützt Cluster davor, versehentlich gelöscht zu werden.

Typ: Boolesch

Erforderlich: Nein

EnableCloudwatchLogsExports.Mitglied.N

Eine Liste von Protokolltypen, die für den Export nach Amazon CloudWatch Logs aktiviert sein müssen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

EngineVersion

Die Version der Datenbank-Engine, die für den neuen Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüssel-ID, die bei der Wiederherstellung eines verschlüsselten Clusters aus einem DB-Snapshot oder Cluster-Snapshot verwendet werden soll.

Die AWS KMS Schlüssel-ID ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben wiederherstellen AWS-Konto , dem der AWS KMS Verschlüsselungsschlüssel gehört, mit dem der neue Cluster verschlüsselt wurde, können Sie den AWS KMS Schlüsselalias anstelle des ARN für den AWS KMS Verschlüsselungsschlüssel verwenden.

Wenn Sie keinen Wert für den `KmsKeyId`-Parameter angeben, geschieht folgendes:

- Wenn der Snapshot oder Cluster-Snapshot verschlüsselt `SnapshotIdentifier` ist, wird der wiederhergestellte Cluster mit dem AWS KMS Schlüssel verschlüsselt, der zum Verschlüsseln des Snapshots oder Cluster-Snapshots verwendet wurde.
- Wenn der Snapshot oder der Cluster-Snapshot nicht verschlüsselt `SnapshotIdentifier` ist, ist der wiederhergestellte DB-Cluster nicht verschlüsselt.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Portnummer, auf der der neue Cluster Verbindungen akzeptiert.

Einschränkungen: Muss ein Wert von `1150` bis sein`65535`.

Standard: Derselbe Port wie der ursprüngliche Cluster.

Typ: Ganzzahl

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Gültige Werte für den Speichertyp - `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem wiederhergestellten Cluster zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

VpcSecurityGroupIds. VpcSecurityGroupIdN

Eine Liste der Virtual Private Cloud (VPC) -Sicherheitsgruppen, zu denen der neue Cluster gehören wird.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterQuotaExceededFault

Der Cluster kann nicht erstellt werden, da Sie das maximal zulässige Kontingent an Clustern erreicht haben.

HTTP Status Code: 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

DBSnapshotNotFound

`DBSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Snapshot.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

InsufficientDBClusterCapacityFault

Der Cluster verfügt nicht über genügend Kapazität für den aktuellen Vorgang.

HTTP Status Code: 403

InsufficientStorageClusterCapacity

Für die aktuelle Aktion ist nicht genügend Speicherplatz verfügbar. Möglicherweise können Sie diesen Fehler beheben, indem Sie Ihre Subnetzgruppe so aktualisieren, dass sie verschiedene Availability Zones verwendet, in denen mehr Speicherplatz verfügbar ist.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBSnapshotState

Der Status des Snapshots erlaubt kein Löschen.

HTTP Status Code: 400

InvalidRestoreFault

Sie können keine Wiederherstellung von einem Virtual Private Cloud (VPC) -Backup auf eine Nicht-VPC-DB-Instance durchführen.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nach ihrer Erstellung nicht alle Availability Zones ab, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen AWS KMS Schlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instanzen verfügbar ist.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instanzen verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RestoreDBClusterToPointInTime

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt einen Cluster zu einem beliebigen Zeitpunkt wieder her. Benutzer können den Zustand jedes beliebigen Zeitpunkts vor `LatestRestorableTime` bis zu `BackupRetentionPeriod` Tagen wiederherstellen. Der Zielcluster wird aus dem Quellcluster mit derselben Konfiguration wie der ursprüngliche Cluster erstellt, mit der Ausnahme, dass der neue Cluster mit der Standardsicherheitsgruppe erstellt wird.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Der Name des neuen Clusters, der erstellt werden soll.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Typ: Zeichenfolge

Erforderlich: Ja

SourceDBClusterIdentifier

Die Kennung des Quell-Clusters, von dem wiederhergestellt werden soll.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen `DBCluster` übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Ja

DBSubnetGroupName

Der Name der Subnetzgruppe, die für den neuen Cluster verwendet werden soll.

Einschränkungen: Falls angegeben, muss er mit dem Namen eines vorhandenen `DBSubnetGroup` übereinstimmen.

Beispiel: `mySubnetgroup`

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn `DeletionProtection` aktiviert, kann der Cluster nur gelöscht werden, wenn er geändert und `DeletionProtection` deaktiviert wurde. `DeletionProtection`schützt Cluster davor, versehentlich gelöscht zu werden.

Typ: Boolesch

Erforderlich: Nein

EnableCloudwatchLogsExports.Mitglied.N

Eine Liste von Protokolltypen, die für den Export nach Amazon CloudWatch Logs aktiviert sein müssen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

KmsKeyId

Die AWS KMS Schlüssel-ID, die bei der Wiederherstellung eines verschlüsselten Clusters aus einem verschlüsselten Cluster verwendet werden soll.

Die AWS KMS Schlüssel-ID ist der Amazon-Ressourcenname (ARN) für den AWS KMS Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben wiederherstellen AWS-Konto , dem der AWS KMS Verschlüsselungsschlüssel gehört, mit dem der neue Cluster verschlüsselt wurde, können Sie den AWS KMS Schlüsselalias anstelle des ARN für den AWS KMS Verschlüsselungsschlüssel verwenden.

Sie können auf einem neuen Cluster wiederherstellen und den neuen Cluster mit einem AWS KMS Schlüssel verschlüsseln, der sich von dem Schlüssel unterscheidet, der AWS KMS zum Verschlüsseln des Quellclusters verwendet wurde. Der neue DB-Cluster wird mit dem durch den Parameter identifizierten AWS KMS Schlüssel verschlüsselt. `KmsKeyId`

Wenn Sie keinen Wert für den `KmsKeyId`-Parameter angeben, geschieht folgendes:

- Wenn der Cluster verschlüsselt ist, wird der wiederhergestellte Cluster mit dem AWS KMS Schlüssel verschlüsselt, der zur Verschlüsselung des Quellclusters verwendet wurde.
- Wenn der Cluster nicht verschlüsselt ist, ist der wiederhergestellte Cluster nicht verschlüsselt.

Wenn es `DBClusterIdentifier` sich auf einen Cluster bezieht, der nicht verschlüsselt ist, wird die Wiederherstellungsanforderung abgelehnt.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Die Portnummer, auf der der neue Cluster Verbindungen akzeptiert.

Einschränkungen: Muss ein Wert von `1150` bis sein`65535`.

Standard: Der Standardanschluss für die Engine.

Typ: Ganzzahl

Erforderlich: Nein

RestoreToTime

Das Datum und die Uhrzeit für die Wiederherstellung des Clusters.

Gültige Werte: Eine Uhrzeit im UTC-Format (Universal Coordinated Time).

Einschränkungen:

- Muss vor dem letzten wiederherstellbaren Zeitpunkt für die Instance liegen.
- Muss angegeben werden, wenn der Parameter `UseLatestRestorableTime` nicht angegeben ist.
- Kann nicht angegeben werden, wenn der Parameter `UseLatestRestorableTime` auf `true` festgelegt ist.
- Kann nicht angegeben werden, wenn der Parameter `RestoreType` auf `copy-on-write` festgelegt ist.

Beispiel: `2015-03-07T23:45:00Z`

Typ: Zeitstempel

Erforderlich: Nein

RestoreType

Der Typ der auszuführenden Wiederherstellung. Sie können einen der folgenden Werte angeben:

- `full-copy` – Der neue DB-Cluster wird als vollständige Kopie des Quell-DB-Clusters wiederhergestellt.
- `copy-on-write` – Der neue DB-Cluster wird als Klon des Quell-DB-Clusters wiederhergestellt.

Einschränkungen: Sie können `copy-on-write` nicht angeben, wenn die Engine-Version des Quell-DB-Clusters älter als 1.11 ist.

Wenn Sie keinen Wert für `RestoreType` angeben, wird der neue DB-Cluster als vollständige Kopie des Quell-DB-Clusters wiederhergestellt.

Typ: Zeichenfolge

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll.

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Gültige Werte für den Speichertyp - `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

Tags.Tag.N

Die Tags, die dem wiederhergestellten Cluster zugewiesen werden sollen.

Typ: Array von Tag-Objekten

Erforderlich: Nein

UseLatestRestorableTime

Ein Wert, der auf `true` festgelegt ist, um den Cluster auf den letzten wiederherstellbaren Sicherungszeitpunkt wiederherzustellen, und sonst `false` anzeigt.

Standard: `false`

Einschränkungen: Darf nicht angegeben werden, wenn der Parameter `RestoreToTime` angegeben ist.

Typ: Boolesch

Erforderlich: Nein

VpcSecurityGroupIds. VpcSecurityGroupIdN

Eine Liste der VPC-Sicherheitsgruppen, zu denen der neue Cluster gehört.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterAlreadyExistsFault

Sie haben bereits einen Cluster mit der angegebenen Kennung.

HTTP Status Code: 400

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

DBClusterQuotaExceededFault

Der Cluster kann nicht erstellt werden, da Sie das maximal zulässige Kontingent an Clustern erreicht haben.

HTTP Status Code: 403

DBClusterSnapshotNotFoundFault

`DBClusterSnapshotIdentifier`bezieht sich nicht auf einen vorhandenen Cluster-Snapshot.

HTTP Status Code: 404

DBSubnetGroupNotFoundFault

`DBSubnetGroupName`bezieht sich nicht auf eine bestehende Subnetzgruppe.

HTTP Status Code: 404

InsufficientDBClusterCapacityFault

Der Cluster verfügt nicht über genügend Kapazität für den aktuellen Vorgang.

HTTP Status Code: 403

InsufficientStorageClusterCapacity

Für die aktuelle Aktion ist nicht genügend Speicherplatz verfügbar. Möglicherweise können Sie diesen Fehler beheben, indem Sie Ihre Subnetzgruppe so aktualisieren, dass sie verschiedene Availability Zones verwendet, in denen mehr Speicherplatz verfügbar ist.

HTTP Status Code: 400

InvalidDBClusterSnapshotStateFault

Der angegebene Wert ist kein gültiger Cluster-Snapshot-Status.

HTTP Status Code: 400

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBSnapshotState

Der Status des Snapshots erlaubt kein Löschen.

HTTP Status Code: 400

InvalidRestoreFault

Sie können keine Wiederherstellung von einem Virtual Private Cloud (VPC) -Backup auf eine Nicht-VPC-DB-Instance durchführen.

HTTP Status Code: 400

InvalidSubnet

Das angeforderte Subnetz ist ungültig, oder es wurden mehrere Subnetze angefordert, die sich nicht alle in einer gemeinsamen Virtual Private Cloud (VPC) befinden.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

Die Subnetzgruppe deckt nach ihrer Erstellung nicht alle Availability Zones ab, da Änderungen vorgenommen wurden.

HTTP Status Code: 400

KMSKeyNotAccessibleFault

Beim Zugriff auf einen AWS KMS Schlüssel ist ein Fehler aufgetreten.

HTTP Status Code: 400

StorageQuotaExceeded

Die Anfrage würde dazu führen, dass Sie die zulässige Menge an Speicherplatz überschreiten, die für alle Instanzen verfügbar ist.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

StartDBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Startet den gestoppten Cluster neu, der von `DBClusterIdentifier` angegeben ist. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster beenden und starten.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die ID des Clusters, der neu gestartet werden soll. Beispiel: `docdb-2019-05-28-15-24-52`

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

StopDBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stoppt den laufenden Cluster, der von angegeben ist`DBClusterIdentifier`. Der Cluster muss sich im Status „Verfügbar“ befinden. Weitere Informationen finden Sie unter Einen Amazon DocumentDB-Cluster beenden und starten.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

DBClusterIdentifier

Die Kennung des Clusters, der gestoppt werden soll. Beispiel: `docdb-2019-05-28-15-24-52`

Typ: Zeichenfolge

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

DBCluster

Detaillierte Informationen zu einem Cluster.

Typ: DBCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidDBInstanceState

Die angegebene Instanz befindet sich nicht im Status „Verfügbar“.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

SwitchoverGlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Schaltet den angegebenen sekundären Amazon DocumentDB-Cluster in den neuen primären Amazon DocumentDB-Cluster im globalen Datenbank-Cluster um.

Anforderungsparameter

Informationen zu den Parametern, die alle Aktionen gemeinsam haben, finden Sie unter Allgemeine Parameter.

GlobalClusterIdentifier

Die Kennung des globalen Amazon DocumentDB DocumentDB-Datenbank-Clusters, zu dem gewechselt werden soll. Die Kennung ist der eindeutige Schlüssel, den der Benutzer bei der Erstellung des Clusters zugewiesen hat. Mit anderen Worten, es ist der Name des globalen Clusters. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss mit der ID eines vorhandenen globalen Clusters (globale Amazon DocumentDB DocumentDB-Datenbank) übereinstimmen.
- Mindestlänge 1. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

TargetDbClusterIdentifier

Die ID des sekundären Amazon DocumentDB-Clusters, der zum neuen primären Cluster für den globalen Datenbank-Cluster heraufgestuft werden soll. Verwenden Sie den Amazon Resource Name (ARN) für die Kennung, damit Amazon DocumentDB den Cluster in seiner AWS Region lokalisieren kann.

Einschränkungen:

- Muss mit der Kennung eines vorhandenen sekundären Clusters übereinstimmen.
- Mindestlänge 1. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Ja

Antwortelemente

Das folgende Element wird vom Dienst zurückgegeben.

GlobalCluster

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Typ: GlobalCluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

DBClusterNotFoundFault

`DBClusterIdentifier`bezieht sich nicht auf einen vorhandenen Cluster.

HTTP Status Code: 404

GlobalClusterNotFoundFault

Das bezieht sich `GlobalClusterIdentifier` nicht auf einen vorhandenen globalen Cluster.

HTTP Status Code: 404

InvalidDBClusterStateFault

Der Cluster befindet sich nicht in einem gültigen Zustand.

HTTP Status Code: 400

InvalidGlobalClusterStateFault

Der angeforderte Vorgang kann nicht ausgeführt werden, solange sich der Cluster in diesem Status befindet.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Amazon DocumentDB Elastic Clusters

Die folgenden Aktionen werden von Amazon DocumentDB Elastic Clusters unterstützt:

- ApplyPendingMaintenanceAction
- CopyClusterSnapshot
- CreateCluster
- CreateClusterSnapshot
- DeleteCluster
- DeleteClusterSnapshot

- GetCluster
- GetClusterSnapshot
- GetPendingMaintenanceAction
- ListClusters
- ListClusterSnapshots
- ListPendingMaintenanceActions
- ListTagsForResource
- RestoreClusterFromSnapshot
- StartCluster
- StopCluster
- TagResource
- UntagResource
- UpdateCluster

ApplyPendingMaintenanceAction

Bedienung: Amazon DocumentDB Elastic Clusters

Der Typ der ausstehenden Wartungsaktion, die auf die Ressource angewendet werden soll.

Anforderungssyntax

```
POST /pending-action HTTP/1.1
Content-type: application/json

{
   "applyAction": "string",
   "applyOn": "string",
   "optInType": "string",
   "resourceArn": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

applyAction

Die ausstehende Wartungsaktion, die auf die Ressource angewendet werden soll.

Gültige Aktionen sind:

- `ENGINE_UPDATE`
- `ENGINE_UPGRADE`
- `SECURITY_UPDATE`
- `OS_UPDATE`
- `MASTER_USER_PASSWORD_UPDATE`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

optInType

Ein Wert, der die Art der Opt-In-Anfrage angibt oder eine Opt-In-Anfrage rückgängig macht. Eine Opt-in-Anfrage vom Typ `IMMEDIATE` kann nicht rückgängig gemacht werden.

Typ: Zeichenfolge

Zulässige Werte: `IMMEDIATE | NEXT_MAINTENANCE | APPLY_ON | UNDO_OPT_IN`

Erforderlich: Ja

resourceArn

Der Amazon DocumentDB Amazon Resource Name (ARN) der Ressource, für die die ausstehende Wartungsaktion gilt.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

applyOn

Ein bestimmtes Datum, an dem die ausstehende Wartungsmaßnahme angewendet werden soll. Erforderlich, wenn opt-in-type ein `APPLY_ON` ist. Format: `yyyy/MM/dd HH:mm-yyyy/MM/dd HH:mm`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "resourcePendingMaintenanceAction": {
      "pendingMaintenanceActionDetails": [
         {
            "action": "string",
            "autoAppliedAfterDate": "string",
            "currentApplyDate": "string",
            "description": "string",
            "forcedApplyDate": "string",
            "optInStatus": "string"
         }
      ],
      "resourceArn": "string"
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

resourcePendingMaintenanceAction

Die Ausgabe der ausstehenden Wartungsaktion, die angewendet wird.

Typ: ResourcePendingMaintenanceAction Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python

- AWS SDK for Ruby V3

CopyClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Kopiert einen Snapshot eines elastischen Clusters.

Anforderungssyntax

```
POST /cluster-snapshot/snapshotArn/copy HTTP/1.1
Content-type: application/json

{
   "copyTags": boolean,
   "kmsKeyId": "string",
   "tags": {
      "string" : "string"
   },
   "targetSnapshotName": "string"
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Der Amazon-Ressourcenname (ARN) -Bezeichner des Elastic Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

targetSnapshotName

Die Kennung des neuen Elastic Cluster-Snapshots, der aus dem Quell-Cluster-Snapshot erstellt werden soll. Bei diesem Parameter wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.

- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `elastic-cluster-snapshot-5`

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 63 Zeichen.

Erforderlich: Ja

copyTags

Wird auf gesetzt, `true` um alle Tags aus dem Quell-Cluster-Snapshot in den Elastic Cluster-Ziel-Snapshot zu kopieren. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

kmsKeyId

Die AWS KMS-Schlüssel-ID für einen verschlüsselten Elastic Cluster-Snapshot. Die AWS KMS-Schlüssel-ID ist der Amazon-Ressourcenname (ARN), die AWS KMS-Schlüssel-ID oder der AWS KMS-Schlüsselalias für den AWS KMS-Verschlüsselungsschlüssel.

Wenn Sie einen verschlüsselten Elastic Cluster-Snapshot von Ihrem AWS Konto kopieren, können Sie einen Wert angeben, `KmsKeyId` um die Kopie mit einem neuen AWS S KMS-Verschlüsselungsschlüssel zu verschlüsseln. Wenn Sie keinen Wert für angeben`KmsKeyId`, wird die Kopie des Elastic Cluster-Snapshots mit demselben `AWS` KMS-Schlüssel verschlüsselt wie der Elastic Cluster-Quell-Snapshot.

Wenn Sie einen unverschlüsselten Elastic Cluster-Snapshot kopieren und einen Wert für den `KmsKeyId` Parameter angeben, wird ein Fehler zurückgegeben.

Typ: Zeichenfolge

Erforderlich: Nein

tags

Die Tags, die dem Elastic Cluster-Snapshot zugewiesen werden sollen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `(?!aws:)[a-zA-Z+-=._:/]+`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen zu einem bestimmten Elastic Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ServiceQuotaExceededException

Das Dienstkontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Drosselung der Anfrage abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateCluster

Service: Amazon DocumentDB Elastic Clusters

Erstellt einen neuen elastischen Amazon DocumentDB-Cluster und gibt dessen Clusterstruktur zurück.

Anforderungssyntax

```
POST /cluster HTTP/1.1
Content-type: application/json

{
   "adminUserName": "string",
   "adminUserPassword": "string",
   "authType": "string",
   "backupRetentionPeriod": number,
   "clientToken": "string",
   "clusterName": "string",
   "kmsKeyId": "string",
   "preferredBackupWindow": "string",
   "preferredMaintenanceWindow": "string",
   "shardCapacity": number,
   "shardCount": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "tags": {
      "string" : "string"
   },
   "vpcSecurityGroupIds": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

adminUserName

Der Name des Amazon DocumentDB Elastic Clusters-Administrators.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben oder Zahlen enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Dies darf kein reserviertes Wort sein.

Typ: Zeichenfolge

Erforderlich: Ja

adminUserPassword

Das Passwort für den Amazon DocumentDB Elastic Clusters-Administrator. Das Passwort kann alle druckbaren ASCII-Zeichen enthalten.

Einschränkungen:

- Muss 8 bis 100 Zeichen enthalten.
- Darf keinen Schrägstrich (/), doppelte Anführungszeichen („) oder das „at“ -Symbol (@) enthalten.

Typ: Zeichenfolge

Erforderlich: Ja

authType

Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das für den Zugriff auf den Elastic Cluster verwendete Passwort abgerufen werden soll. Gültige Typen sind `PLAIN_TEXT` oder`SECRET_ARN`.

Typ: Zeichenfolge

Zulässige Werte: `PLAIN_TEXT | SECRET_ARN`

Erforderlich: Ja

clusterName

Der Name des neuen elastischen Clusters. Dieser Parameter wird als Zeichenfolge in Kleinbuchstaben gespeichert.

Einschränkungen:

- Muss zwischen 1 und 63 Buchstaben, Ziffern oder Bindestriche enthalten.

- Das erste Zeichen muss ein Buchstabe sein.
- Darf nicht mit einem Bindestrich enden oder zwei aufeinanderfolgende Bindestriche enthalten.

Beispiel: `my-cluster`

Typ: Zeichenfolge

Erforderlich: Ja

shardCapacity

Die Anzahl von v, die jedem Elastic Cluster-Shard CPUs zugewiesen wurden. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.

Typ: Ganzzahl

Erforderlich: Ja

shardCount

Die Anzahl der Shards, die dem elastischen Cluster zugewiesen sind. Das Maximum ist 32.

Typ: Ganzzahl

Erforderlich: Ja

backupRetentionPeriod

Die Anzahl der Tage, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

clientToken

Das Client-Token für den Elastic Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

kmsKeyId

Die KMS-Schlüssel-ID, die zum Verschlüsseln des neuen Elastic Clusters verwendet werden soll.

Die Kennung für den KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon-Konto erstellen, dem dieser KMS-Verschlüsselungsschlüssel gehört, können Sie den KMS-Schlüsselalias anstelle des ARN als KMS-Verschlüsselungsschlüssel verwenden.

Wenn kein Verschlüsselungsschlüssel angegeben ist, verwendet Amazon DocumentDB den Standardverschlüsselungsschlüssel, den KMS für Ihr Konto erstellt. Ihr Konto hat für jede Amazon-Region einen anderen Standard-Verschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

preferredBackupWindow

Der tägliche Zeitraum, in dem automatische Backups erstellt werden, wenn automatische Backups aktiviert sind, wie von der festgelegt`backupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

preferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Standard: Ein 30-Minuten-Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils AWS-Region 8 Stunden ausgewählt wird und an einem zufälligen Wochentag stattfindet.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikatinstanzen, die für alle Shards im Elastic Cluster gelten. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt, und alle

weiteren Instances sind Replikate, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

subnetIds

Das EC2 Amazon-Subnetz IDs für den neuen Elastic Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

tags

Die Tags, die dem neuen elastischen Cluster zugewiesen werden sollen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `(?!aws:)[a-zA-Z+-=._:/]+`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

vpcSecurityGroupIds

Eine Liste von EC2 VPC-Sicherheitsgruppen, die dem neuen elastischen Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Der neue elastische Cluster, der erstellt wurde.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ServiceQuotaExceededException

Das Dienstkontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Drosselung der Anfrage abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

CreateClusterSnapshot

Service: Amazon DocumentDB Elastic Clusters

Erstellt einen Snapshot eines elastischen Clusters.

Anforderungssyntax

```
POST /cluster-snapshot HTTP/1.1
Content-type: application/json

{
   "clusterArn": "string",
   "snapshotName": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI-Anfrageparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

clusterArn

Die ARN-ID des Elastic Clusters, von dem Sie einen Snapshot erstellen möchten.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotName

Der Name des neuen Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 63 Zeichen.

Erforderlich: Ja

tags

Die Tags, die dem neuen Elastic Cluster-Snapshot zugewiesen werden sollen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `(?!aws:)[a-zA-Z+-=._:/]+`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen über den neuen Elastic Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ServiceQuotaExceededException

Das Dienstkontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Drosselung der Anfrage abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteCluster

Bedienung: Amazon DocumentDB Elastic Clusters

Löschen Sie einen elastischen Cluster.

Anforderungssyntax

```
DELETE /cluster/clusterArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters, der gelöscht werden soll.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über den neu gelöschten Elastic Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

DeleteClusterSnapshot

Bedienung: Amazon DocumentDB Elastic Clusters

Löschen Sie einen Elastic Cluster-Snapshot.

Anforderungssyntax

```
DELETE /cluster-snapshot/snapshotArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die ARN-ID des Elastic Cluster-Snapshots, der gelöscht werden soll.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
```

```
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen über den neu gelöschten Elastic Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetCluster

Bedienung: Amazon DocumentDB Elastic Clusters

Gibt Informationen über einen bestimmten elastischen Cluster zurück.

Anforderungssyntax

```
GET /cluster/clusterArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über einen bestimmten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetClusterSnapshot

Bedienung: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu einem bestimmten Elastic Cluster-Snapshot zurück

Anforderungssyntax

```
GET /cluster-snapshot/snapshotArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die ARN-ID des Elastic Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "snapshot": {
      "adminUserName": "string",
      "clusterArn": "string",
      "clusterCreationTime": "string",
      "kmsKeyId": "string",
      "snapshotArn": "string",
      "snapshotCreationTime": "string",
      "snapshotName": "string",
      "snapshotType": "string",
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
```

```
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

snapshot

Gibt Informationen zu einem bestimmten Elastic Cluster-Snapshot zurück.

Typ: ClusterSnapshot Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

GetPendingMaintenanceAction

Bedienung: Amazon DocumentDB Elastic Clusters

Ruft alle ausstehenden Wartungsaktionen ab.

Anforderungssyntax

```
GET /pending-action/resourceArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Ruft ausstehende Wartungsaktionen für einen bestimmten Amazon-Ressourcennamen (ARN) ab.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "resourcePendingMaintenanceAction": {
      "pendingMaintenanceActionDetails": [
         {
            "action": "string",
            "autoAppliedAfterDate": "string",
            "currentApplyDate": "string",
            "description": "string",
            "forcedApplyDate": "string",
            "optInStatus": "string"
         }
```

```
        ],
        "resourceArn": "string"
    }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

resourcePendingMaintenanceAction

Stellt Informationen über eine ausstehende Wartungsaktion für eine Ressource bereit.

Typ: ResourcePendingMaintenanceAction Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zur Durchführung einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ListClusters

Bedienung: Amazon DocumentDB Elastic Clusters

Gibt Informationen über bereitgestellte elastische Amazon DocumentDB-Cluster zurück.

Anforderungssyntax

```
GET /clusters?maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

maxResults

Die maximale Anzahl von Elastic Cluster-Snapshot-Ergebnissen, die in der Antwort empfangen werden sollen.

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

nextToken

Ein Paginierungstoken, das durch eine vorherige Anfrage bereitgestellt wurde. Wenn dieser Parameter angegeben ist, umfasst die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von `max-results` angegebenen Wert.

Wenn die Antwort keine weiteren Daten enthält, `nextToken` werden sie nicht zurückgegeben.

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "clusters": [
      {
         "clusterArn": "string",
         "clusterName": "string",
```

```
            "status": "string"
        }
    ],
    "nextToken": "string"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

clusters

Eine Liste elastischer Amazon DocumentDB-Cluster.

Typ: Array von ClusterInList-Objekten

nextToken

Ein Paginierungstoken, das durch eine vorherige Anfrage bereitgestellt wurde. Wenn dieser Parameter angegeben ist, umfasst die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von `max-results` angegebenen Wert.

Wenn die Antwort keine weiteren Daten enthält, `nextToken` werden sie nicht zurückgegeben.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ListClusterSnapshots

Bedienung: Amazon DocumentDB Elastic Clusters

Gibt Informationen über Snapshots für einen angegebenen Elastic Cluster zurück.

Anforderungssyntax

```
GET /cluster-snapshots?
clusterArn=clusterArn&maxResults=maxResults&nextToken=nextToken&snapshotType=snapshotType
 HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

maxResults

Die maximale Anzahl von Elastic Cluster-Snapshot-Ergebnissen, die in der Antwort empfangen werden sollen.

Gültiger Bereich: Mindestwert von 20. Maximalwert 100.

nextToken

Ein Paginierungstoken, das durch eine vorherige Anfrage bereitgestellt wurde. Wenn dieser Parameter angegeben ist, umfasst die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von `max-results` angegebenen Wert.

Wenn die Antwort keine weiteren Daten enthält, `nextToken` werden sie nicht zurückgegeben.

snapshotType

Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:

- `automated`- Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr AWS Konto erstellt hat.
- `manual`- Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr Konto erstellt haben. AWS

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "snapshots": [
      {
         "clusterArn": "string",
         "snapshotArn": "string",
         "snapshotCreationTime": "string",
         "snapshotName": "string",
         "status": "string"
      }
   ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

nextToken

Ein Paginierungstoken, das durch eine vorherige Anfrage bereitgestellt wurde. Wenn dieser Parameter angegeben ist, umfasst die Antwort nur Datensätze, die über dieses Token hinausgehen, bis zu dem von `max-results` angegebenen Wert.

Wenn die Antwort keine weiteren Daten enthält, `nextToken` werden sie nicht zurückgegeben.

Typ: Zeichenfolge

snapshots

Eine Liste von Snapshots für einen bestimmten Elastic Cluster.

Typ: Array von ClusterSnapshotInList-Objekten

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3

- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ListPendingMaintenanceActions

Bedienung: Amazon DocumentDB Elastic Clusters

Ruft eine Liste aller ausstehenden Wartungsaktionen ab.

Anforderungssyntax

```
GET /pending-actions?maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

maxResults

Die maximale Anzahl von Ergebnissen, die in die Antwort aufgenommen werden sollen. Wenn mehr Datensätze als der angegebene `maxResults` Wert vorhanden sind, ist ein Paginierungstoken (Markierung) in der Antwort enthalten, sodass die verbleibenden Ergebnisse abgerufen werden können.

Gültiger Bereich: Mindestwert 1. Maximalwert 100.

nextToken

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `maxResults` angegebenen Wert.

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "nextToken": "string",
   "resourcePendingMaintenanceActions": [
      {
         "pendingMaintenanceActionDetails": [
```

```
            {
               "action": "string",
               "autoAppliedAfterDate": "string",
               "currentApplyDate": "string",
               "description": "string",
               "forcedApplyDate": "string",
               "optInStatus": "string"
            }
         ],
         "resourceArn": "string"
      }
   ]
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

resourcePendingMaintenanceActions

Stellt Informationen über eine ausstehende Wartungsaktion für eine Ressource bereit.

Typ: Array von ResourcePendingMaintenanceAction-Objekten

nextToken

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn dieser Parameter angezeigt wird, enthalten die Antworten nur Datensätze, die sich hinter der Markierung befinden, bis zu dem von `maxResults` angegebenen Wert.

Typ: Zeichenfolge

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

ListTagsForResource

Service: Amazon DocumentDB Elastic Clusters

Listet alle Tags auf einer Elastic Cluster-Ressource auf

Anforderungssyntax

```
GET /tags/resourceArn HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Die ARN-ID der Elastic Cluster-Ressource.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Die maximale Länge beträgt 1011.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

tags

Die Liste der Tags für die angegebene Elastic Cluster-Ressource.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `(?!aws:)[a-zA-Z+-=._:/]+`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

RestoreClusterFromSnapshot

Service: Amazon DocumentDB Elastic Clusters

Stellt einen elastischen Cluster aus einem Snapshot wieder her.

Anforderungssyntax

```
POST /cluster-snapshot/snapshotArn/restore HTTP/1.1
Content-type: application/json

{
   "clusterName": "string",
   "kmsKeyId": "string",
   "shardCapacity": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "tags": {
      "string" : "string"
   },
   "vpcSecurityGroupIds": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

snapshotArn

Die ARN-ID des Elastic Cluster-Snapshots.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

clusterName

Der Name des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

kmsKeyId

Die KMS-Schlüssel-ID, die zur Verschlüsselung des neuen Amazon DocumentDB DocumentDB-Elastic-Clusters-Clusters verwendet werden soll.

Die Kennung für den KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon-Konto erstellen, dem dieser KMS-Verschlüsselungsschlüssel gehört, können Sie den KMS-Schlüsselalias anstelle des ARN als KMS-Verschlüsselungsschlüssel verwenden.

Wenn hier kein Verschlüsselungsschlüssel angegeben ist, verwendet Amazon DocumentDB den Standardverschlüsselungsschlüssel, den KMS für Ihr Konto erstellt. Ihr Konto hat für jede Amazon-Region einen anderen Standard-Verschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

shardCapacity

Die Kapazität jedes Shards im neuen wiederhergestellten Elastic Cluster.

Typ: Ganzzahl

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikatinstanzen, die für alle Shards im Elastic Cluster gelten. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt, und alle weiteren Instances sind Replikate, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

subnetIds

Das EC2 Amazon-Subnetz IDs für den Elastic Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

tags

Eine Liste der Tag-Namen, die dem wiederhergestellten Elastic Cluster zugewiesen werden sollen, in Form einer Reihe von Schlüssel-Wert-Paaren, wobei der Schlüssel der Tag-Name und der Wert der Schlüsselwert ist.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `(?!aws:)[a-zA-Z+-=._:/]+`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

vpcSecurityGroupIds

Eine Liste von EC2 VPC-Sicherheitsgruppen, die dem Elastic Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über den wiederhergestellten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ServiceQuotaExceededException

Das Dienstkontingent für die Aktion wurde überschritten.

HTTP-Statuscode: 402

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Drosselung der Anfrage abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin

- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

StartCluster

Bedienung: Amazon DocumentDB Elastic Clusters

Startet den gestoppten Elastic Cluster neu, der von `clusterARN` angegeben ist.

Anforderungssyntax

```
POST /cluster/clusterArn/start HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
```

```
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über einen bestimmten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

StopCluster

Bedienung: Amazon DocumentDB Elastic Clusters

Stoppt den laufenden Elastic Cluster, der von angegeben ist`clusterArn`. Der elastische Cluster muss sich im Status „Verfügbar“ befinden.

Anforderungssyntax

```
POST /cluster/clusterArn/stop HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Die ARN-ID des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
```

```
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über einen bestimmten elastischen Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

TagResource

Service: Amazon DocumentDB Elastic Clusters

Fügt einer Elastic Cluster-Ressource Metadaten-Tags hinzu

Anforderungssyntax

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Die ARN-ID der Elastic Cluster-Ressource.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Die maximale Länge beträgt 1011.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

tags

Die Tags, die der Elastic Cluster-Ressource zugewiesen sind.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Schlüssel-Längenbeschränkungen: Minimale Länge von 1. Maximale Länge beträgt 128 Zeichen.

Schlüssel-Muster: `(?!aws:)[a-zA-Z+-=._:/]+`

Längenbeschränkungen für Werte: Mindestlänge von 0. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

UntagResource

Service: Amazon DocumentDB Elastic Clusters

Entfernt Metadaten-Tags aus einer Elastic Cluster-Ressource

Anforderungssyntax

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

resourceArn

Die ARN-ID der Elastic Cluster-Ressource.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Die maximale Länge beträgt 1011.

Erforderlich: Ja

tagKeys

Die Tag-Schlüssel, die aus der Elastic Cluster-Ressource entfernt werden sollen.

Array-Mitglieder: Die Mindestanzahl beträgt 0 Elemente. Die maximale Anzahl beträgt 50 Elemente.

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 128 Zeichen.

Pattern: `(?!aws:)[a-zA-Z+-=._:/]+`

Erforderlich: Ja

Anforderungstext

Der Anforderung besitzt keinen Anforderungstext.

Antwortsyntax

```
HTTP/1.1 200
```

Antwortelemente

Wenn die Aktion erfolgreich ist, gibt der Dienst eine HTTP 200-Antwort mit leerem HTTP-Textinhalt zurück.

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

UpdateCluster

Bedienung: Amazon DocumentDB Elastic Clusters

Ändert einen elastischen Cluster. Dazu gehören die Aktualisierung von Admin-Benutzername/Passwort, die Aktualisierung der API-Version und die Einrichtung eines Backup-Fensters und eines Wartungsfensters

Anforderungssyntax

```
PUT /cluster/clusterArn HTTP/1.1
Content-type: application/json

{
   "adminUserPassword": "string",
   "authType": "string",
   "backupRetentionPeriod": number,
   "clientToken": "string",
   "preferredBackupWindow": "string",
   "preferredMaintenanceWindow": "string",
   "shardCapacity": number,
   "shardCount": number,
   "shardInstanceCount": number,
   "subnetIds": [ "string" ],
   "vpcSecurityGroupIds": [ "string" ]
}
```

URI-Anfrageparameter

Die Anforderung verwendet die folgenden URI-Parameter.

clusterArn

Der ARN-Bezeichner des elastischen Clusters.

Erforderlich: Ja

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

adminUserPassword

Das dem Elastic Cluster-Administrator zugeordnete Passwort. Dieses Passwort kann alle druckbaren ASCII-Zeichen, außer Schrägstrich (/), doppeltes Anführungszeichen (") oder das "At"-Zeichen (@), enthalten.

Einschränkungen: Muss zwischen 8 und 100 Zeichen enthalten.

Typ: Zeichenfolge

Erforderlich: Nein

authType

Der Authentifizierungstyp, der verwendet wird, um zu bestimmen, wo das Passwort abgerufen werden soll, das für den Zugriff auf den Elastic Cluster verwendet wird. Gültige Typen sind `PLAIN_TEXT` oder`SECRET_ARN`.

Typ: Zeichenfolge

Zulässige Werte: `PLAIN_TEXT | SECRET_ARN`

Erforderlich: Nein

backupRetentionPeriod

Die Anzahl der Tage, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

clientToken

Das Client-Token für den Elastic Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

preferredBackupWindow

Der tägliche Zeitraum, in dem automatische Backups erstellt werden, wenn automatische Backups aktiviert sind, wie durch den bestimmt`backupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

preferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Standard: Ein 30-Minuten-Fenster, das nach dem Zufallsprinzip aus einem Zeitblock von jeweils AWS-Region 8 Stunden ausgewählt wird und an einem zufälligen Wochentag stattfindet.

Gültige Tage: Mo, Di, Mi, Do, Fr, Sa, So

Einschränkungen: mindestens 30-Minuten-Zeitfenster.

Typ: Zeichenfolge

Erforderlich: Nein

shardCapacity

Die Anzahl von v, die jedem elastischen Cluster-Shard CPUs zugewiesen ist. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.

Typ: Ganzzahl

Erforderlich: Nein

shardCount

Die Anzahl der Shards, die dem elastischen Cluster zugewiesen sind. Das Maximum ist 32.

Typ: Ganzzahl

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikatinstanzen, die für alle Shards im Elastic Cluster gelten. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instance gibt, und alle weiteren Instances sind Replikate, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

subnetIds

Das EC2 Amazon-Subnetz IDs für den Elastic Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

vpcSecurityGroupIds

Eine Liste von EC2 VPC-Sicherheitsgruppen, die dem Elastic Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json

{
   "cluster": {
      "adminUserName": "string",
      "authType": "string",
      "backupRetentionPeriod": number,
      "clusterArn": "string",
      "clusterEndpoint": "string",
      "clusterName": "string",
      "createTime": "string",
      "kmsKeyId": "string",
      "preferredBackupWindow": "string",
      "preferredMaintenanceWindow": "string",
      "shardCapacity": number,
      "shardCount": number,
      "shardInstanceCount": number,
      "shards": [
         {
            "createTime": "string",
            "shardId": "string",
```

```
            "status": "string"
         }
      ],
      "status": "string",
      "subnetIds": [ "string" ],
      "vpcSecurityGroupIds": [ "string" ]
   }
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden vom Service im JSON-Format zurückgegeben.

cluster

Gibt Informationen über den aktualisierten Elastic Cluster zurück.

Typ: Cluster Objekt

Fehler

Weitere Informationen zu den allgemeinen Fehlern, die bei allen Aktionen zurückgegeben werden, finden Sie unter Häufige Fehler.

AccessDeniedException

Eine Ausnahme, die auftritt, wenn nicht genügend Berechtigungen zum Ausführen einer Aktion vorhanden sind.

HTTP Status Code: 403

ConflictException

Es gab einen Zugriffskonflikt.

HTTP-Statuscode: 409

InternalServerException

Es ist ein interner Serverfehler aufgetreten.

HTTP Status Code: 500

ResourceNotFoundException

Die angegebene Ressource konnte nicht gefunden werden.

HTTP Status Code: 404

ThrottlingException

ThrottlingException wird ausgelöst, wenn die Anfrage aufgrund einer Anforderungsdrosselung abgelehnt wurde.

HTTP-Statuscode: 429

ValidationException

Eine Struktur, die eine Validierungsausnahme definiert.

HTTP Status Code: 400

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS -Befehlszeilenschnittstelle
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK für JavaScript V3
- AWS SDK für Kotlin
- AWS SDK for PHP V3
- AWS SDK für Python
- AWS SDK for Ruby V3

Datentypen

Die folgenden Datentypen werden unterstützt von Amazon DocumentDB (with MongoDB compatibility):

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- ClusterMasterUserSecret
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute
- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions

- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

Die folgenden Datentypen werden von Amazon DocumentDB Elastic Clusters unterstützt:

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- PendingMaintenanceActionDetails
- ResourcePendingMaintenanceAction
- Shard
- ValidationExceptionField

Amazon DocumentDB (with MongoDB compatibility)

Die folgenden Datentypen werden unterstützt von Amazon DocumentDB (with MongoDB compatibility):

- AvailabilityZone
- Certificate
- CertificateDetails
- CloudwatchLogsExportConfiguration
- ClusterMasterUserSecret
- DBCluster
- DBClusterMember
- DBClusterParameterGroup
- DBClusterRole
- DBClusterSnapshot
- DBClusterSnapshotAttribute

- DBClusterSnapshotAttributesResult
- DBEngineVersion
- DBInstance
- DBInstanceStatusInfo
- DBSubnetGroup
- Endpoint
- EngineDefaults
- Event
- EventCategoriesMap
- EventSubscription
- Filter
- GlobalCluster
- GlobalClusterMember
- OrderableDBInstanceOption
- Parameter
- PendingCloudwatchLogsExports
- PendingMaintenanceAction
- PendingModifiedValues
- ResourcePendingMaintenanceActions
- Subnet
- Tag
- UpgradeTarget
- VpcSecurityGroupMembership

AvailabilityZone

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Informationen zu einer Availability Zone.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

Name

Der Name der Availability Zone.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Certificate

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein Zertifikat der Zertifizierungsstelle (CA) für ein AWS-Konto.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

CertificateArn

Der Amazon-Ressourcenname (ARN) für das Zertifikat.

Beispiel: `arn:aws:rds:us-east-1::cert:rds-ca-2019`

Typ: Zeichenfolge

Erforderlich: Nein

CertificateIdentifier

Der eindeutige Schlüssel, der ein Zertifikat identifiziert.

Beispiel: `rds-ca-2019`

Typ: Zeichenfolge

Erforderlich: Nein

CertificateType

Der Typ des Zertifikats.

Beispiel: `CA`

Typ: Zeichenfolge

Erforderlich: Nein

Thumbprint

Der Fingerabdruck des Zertifikats.

Typ: Zeichenfolge

Erforderlich: Nein

ValidFrom

Das Startdatum und die Uhrzeit, ab der das Zertifikat gültig ist.

Beispiel: `2019-07-31T17:57:09Z`

Typ: Zeitstempel

Erforderlich: Nein

ValidTill

Das Datum und die Uhrzeit, nach der das Zertifikat nicht mehr gültig ist.

Beispiel: `2024-07-31T17:57:09Z`

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CertificateDetails

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Gibt die Details des Serverzertifikats der DB-Instance zurück.

Weitere Informationen finden Sie unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten bei der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

CAIdentifier

Die CA-ID des CA-Zertifikats, das für das Serverzertifikat der DB-Instance verwendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

ValidTill

Das Ablaufdatum des Serverzertifikats der DB-Instance.

Typ: Zeitstempel

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CloudwatchLogsExportConfiguration

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Die Konfigurationseinstellung für die Protokolltypen, die für den Export nach Amazon CloudWatch Logs für eine bestimmte Instance oder einen bestimmten Cluster aktiviert werden sollen.

Die `DisableLogTypes` Arrays `EnableLogTypes` und bestimmen, welche Protokolle in Logs exportiert (oder nicht exportiert) werden CloudWatch . Die Werte in diesen Arrays hängen von der verwendeten Engine ab.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DisableLogTypes.member.N

Die Liste der Protokolltypen, die deaktiviert werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

EnableLogTypes.member.N

Die Liste der Protokolltypen, die aktiviert werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterMasterUserSecret

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält das von Amazon DocumentDB in AWS Secrets Manager verwaltete Geheimnis für das Masterbenutzerkennwort.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

KmsKeyId

Die AWS KMS-Schlüssel-ID, die zur Verschlüsselung des Geheimnisses verwendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

SecretArn

Der Amazon-Ressourcenname (ARN) des Secrets.

Typ: Zeichenfolge

Erforderlich: Nein

SecretStatus

Der Status des Geheimnisses.

Mögliche Werte für den Status sind u. a. folgende:

- erstellen — Das Geheimnis wird gerade erstellt.
- aktiv — Das Geheimnis ist für den normalen Gebrauch und die Rotation verfügbar.
- rotierend — Das Geheimnis wird rotiert.
- beeinträchtigt — Das Geheimnis kann für den Zugriff auf Datenbankanmeldedaten verwendet werden, es kann jedoch nicht rotiert werden. Ein Geheimnis kann diesen Status haben, wenn beispielsweise die Berechtigungen geändert werden, sodass Amazon DocumentDB weder auf das Geheimnis noch auf den KMS-Schlüssel für das Geheimnis mehr zugreifen kann.

Wenn ein Secret diesen Status hat, können Sie die Bedingung korrigieren, die den Status verursacht hat. Alternativ können Sie die Instance ändern, um die automatische Verwaltung von Datenbankanmeldedaten zu deaktivieren, und dann die Instance erneut ändern, um die automatische Verwaltung von Datenbankanmeldedaten zu aktivieren.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem Cluster.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

AssociatedRoles.DBClusterRole.N

Stellt eine Liste der AWS Identity and Access Management (IAM-) Rollen bereit, die dem Cluster zugeordnet sind. (IAM-) Rollen, die einem Cluster zugeordnet sind, gewähren dem Cluster die Berechtigung, in Ihrem Namen auf andere AWS Dienste zuzugreifen.

Typ: Array von DBClusterRole-Objekten

Erforderlich: Nein

AvailabilityZones.AvailabilityZone.N

Stellt die Liste der Amazon EC2 Availability Zones bereit, in denen Instances im Cluster erstellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

BackupRetentionPeriod

Gibt die Anzahl der Tage an, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

CloneGroupId

Identifiziert die Clone-Gruppe, mit der der DB-Cluster verknüpft ist.

Typ: Zeichenfolge

Erforderlich: Nein

ClusterCreateTime

Gibt die Uhrzeit der Clustererstellung in UTC (Universal Coordinated Time) an.

Typ: Zeitstempel

Erforderlich: Nein

DBClusterArn

Der Amazon-Ressourcenname (ARN) für den Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterIdentifier

Enthält eine vom Benutzer angegebene Cluster-ID. Dieser Bezeichner ist der eindeutige Schlüssel, der einen Cluster identifiziert.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterMembers.DBClusterMember.N

Stellt die Liste der Instanzen bereit, aus denen der Cluster besteht.

Typ: Array von DBClusterMember-Objekten

Erforderlich: Nein

DBClusterParameterGroup

Gibt den Namen der Cluster-Parametergruppe für den Cluster an.

Typ: Zeichenfolge

Erforderlich: Nein

DbClusterResourceId

Der AWS-Region eindeutige, unveränderliche Bezeichner für den Cluster. Dieser Bezeichner wird in AWS CloudTrail Protokolleinträgen immer dann gefunden, wenn auf den AWS KMS Schlüssel für den Cluster zugegriffen wird.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroup

Gibt Informationen über die Subnetzgruppe an, die dem Cluster zugeordnet ist, einschließlich des Namens, der Beschreibung und der Subnetze in der Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Gibt an, ob dieser Cluster gelöscht werden kann. Wenn `DeletionProtection` aktiviert, kann der Cluster nur gelöscht werden, wenn er geändert und `DeletionProtection` deaktiviert wurde. `DeletionProtection`schützt Cluster davor, versehentlich gelöscht zu werden.

Typ: Boolesch

Erforderlich: Nein

EarliestRestorableTime

Der früheste Zeitpunkt, zu dem eine Datenbank mit point-in-time Restore wiederhergestellt werden kann.

Typ: Zeitstempel

Erforderlich: Nein

EnabledCloudwatchLogsExports.member.N

Eine Liste der Protokolltypen, für deren Export dieser Cluster nach Amazon CloudWatch Logs konfiguriert ist.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Endpoint

Gibt den Verbindungsendpunkt für die primäre Instance des Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Gibt den Namen der Datenbank-Engine an, die für diesen Cluster verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Erforderlich: Nein

HostedZoneId

Gibt die ID an, die Amazon Route 53 zuweist, wenn Sie eine gehostete Zone erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

KmsKeyId

Falls `StorageEncrypted` j`true`, der AWS KMS Schlüsselbezeichner für den verschlüsselten Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

LatestRestorableTime

Gibt den letzten Zeitpunkt an, zu dem eine Datenbank mit point-in-time restore wiederhergestellt werden kann.

Typ: Zeitstempel

Erforderlich: Nein

MasterUsername

Enthält den Master-Benutzernamen für den Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserSecret

Das von Amazon DocumentDB in AWS Secrets Manager verwaltete Geheimnis für das Masterbenutzer-Passwort.

Typ: ClusterMasterUserSecret Objekt

Erforderlich: Nein

MultiAZ

Gibt an, ob der Cluster über Instances in mehreren Availability Zones verfügt.

Typ: Boolesch

Erforderlich: Nein

PercentProgress

Gibt den Fortschritt der Operation als Prozentsatz an.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Gibt die Portnummer an, die von der Datenbank-Engine überwacht wird.

Typ: Ganzzahl

Erforderlich: Nein

PreferredBackupWindow

Gibt den täglichen Zeitraum in koordinierter Weltzeit (UTC) an, in dem automatische Sicherungen erstellt werden, wenn automatische Sicherungen aktiviert sind, gemäß `BackupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Gibt den wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time) an.

Typ: Zeichenfolge

Erforderlich: Nein

ReaderEndpoint

Der Leser-Endpunkt für den Cluster. Der Reader-Endpunkt für einen Cluster verteilt die Lasten der Verbindungen zwischen den Amazon DocumentDB DocumentDB-Replikaten, die in einem Cluster verfügbar sind. Wenn Clients neue Verbindungen zum Reader-Endpunkt anfordern, verteilt Amazon DocumentDB die Verbindungsanfragen auf die Amazon DocumentDB DocumentDB-Replikate im Cluster. Diese Funktion kann Ihnen helfen, Ihre Lese-Arbeitslast auf mehrere Amazon DocumentDB DocumentDB-Replikate in Ihrem Cluster zu verteilen.

Wenn ein Failover auftritt und das Amazon DocumentDB DocumentDB-Replikat, mit dem Sie verbunden sind, zur primären Instance hochgestuft wird, wird Ihre Verbindung unterbrochen. Um Ihren Lese-Workload weiterhin an andere Amazon DocumentDB DocumentDB-Replikate im Cluster zu senden, können Sie dann erneut eine Verbindung zum Reader-Endpunkt herstellen.

Typ: Zeichenfolge

Erforderlich: Nein

ReadReplicaIdentifiers.ReadReplicaIdentifier.N

Enthält einen oder mehrere Identifikatoren der sekundären Cluster, die diesem Cluster zugeordnet sind.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

ReplicationSourceIdentifier

Enthält den Bezeichner des Quellclusters, wenn es sich bei diesem Cluster um einen sekundären Cluster handelt.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Gibt den aktuellen Status dieses Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Gibt an, ob der Cluster verschlüsselt ist.

Typ: Boolesch

Erforderlich: Nein

StorageType

Speichertyp, der Ihrem Cluster zugeordnet ist

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Gültige Werte für den Speichertyp - `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

VpcSecurityGroups.VpcSecurityGroupMembership.N

Stellt eine Liste der Virtual Private Cloud (VPC) -Sicherheitsgruppen bereit, zu denen der Cluster gehört.

Typ: Array von VpcSecurityGroupMembership-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterMember

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält Informationen über eine Instanz, die Teil eines Clusters ist.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterParameterGroupStatus

Gibt den Status der Cluster-Parametergruppe für dieses Mitglied des DB-Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceIdentifier

Gibt die Instance-ID für dieses Mitglied des Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

IsClusterWriter

Ein Wert, der angibt, `true` ob das Cluster-Mitglied die primäre Instanz für den Cluster ist und `false` andernfalls.

Typ: Boolesch

Erforderlich: Nein

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance heraufgestuft wird.

Typ: Ganzzahl

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterParameterGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Cluster-Parametergruppe.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterParameterGroupArn

Der Amazon-Ressourcenname (ARN) für die Cluster-Parametergruppe.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterParameterGroupName

Stellt den Namen der Cluster-Parametergruppe bereit.

Typ: Zeichenfolge

Erforderlich: Nein

DBParameterGroupFamily

Gibt den Namen der Parametergruppenfamilie an, mit der diese Cluster-Parametergruppe kompatibel ist.

Typ: Zeichenfolge

Erforderlich: Nein

Description

Stellt die vom Kunden angegebene Beschreibung für diese Cluster-Parametergruppe bereit.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterRole

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Beschreibt eine AWS Identity and Access Management (IAM-) Rolle, die einem Cluster zugeordnet ist.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

RoleArn

Der Amazon-Ressourcenname (ARN) des IAMrole , der dem DB-Cluster zugeordnet ist.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Beschreibt den Status der Zuordnung zwischen dem IAMrole und dem Cluster. Die `Status` Eigenschaft gibt einen der folgenden Werte zurück:

- `ACTIVE`- Der IAMrole ARN ist mit dem Cluster verknüpft und kann verwendet werden, um in Ihrem Namen auf andere AWS Dienste zuzugreifen.
- `PENDING`- Der IAMrole ARN wird mit dem Cluster verknüpft.
- `INVALID`- Der IAMrole ARN ist mit dem Cluster verknüpft, aber der Cluster kann nicht davon ausgehen, dass er in Ihrem Namen auf andere AWS Dienste zugreift. IAMrole

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterSnapshot

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem Cluster-Snapshot.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AvailabilityZones.AvailabilityZone.N

Stellt die Liste der Amazon EC2 Availability Zones bereit, in denen Instances im Cluster-Snapshot wiederhergestellt werden können.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

ClusterCreateTime

Gibt die Uhrzeit der Erstellung des Clusters in UTC (Universal Coordinated Time) an.

Typ: Zeitstempel

Erforderlich: Nein

DBClusterIdentifier

Gibt die Cluster-ID des Clusters an, aus dem dieser Cluster-Snapshot erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterSnapshotArn

Der Amazon-Ressourcenname (ARN) für den Cluster-Snapshot.

Typ: Zeichenfolge

Erforderlich: Nein

DBClusterSnapshotIdentifier

Gibt den Bezeichner für den Cluster-Snapshot an.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Legt den Namen der Datenbank-Engine fest.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Stellt die Version der Datenbank-Engine für diesen Cluster-Snapshot bereit.

Typ: Zeichenfolge

Erforderlich: Nein

KmsKeyId

Falls `StorageEncrypted` j`atrue`, die AWS KMS Schlüssel-ID für den verschlüsselten Cluster-Snapshot.

Typ: Zeichenfolge

Erforderlich: Nein

MasterUsername

Stellt den Master-Benutzernamen für den Cluster-Snapshot bereit.

Typ: Zeichenfolge

Erforderlich: Nein

PercentProgress

Gibt einen Prozentsatz der Daten an, die laut Schätzung bereits übertragen wurden.

Typ: Ganzzahl

Erforderlich: Nein

Port

Gibt den Port an, auf dem der Cluster zum Zeitpunkt des Snapshots gewartet hat.

Typ: Ganzzahl

Erforderlich: Nein

SnapshotCreateTime

Gibt die Uhrzeit der Erstellung des Snapshots in UTC an.

Typ: Zeitstempel

Erforderlich: Nein

SnapshotType

Gibt den Typ des Cluster-Snapshots an.

Typ: Zeichenfolge

Erforderlich: Nein

SourceDBClusterSnapshotArn

Wenn der Cluster-Snapshot aus einem Quell-Cluster-Snapshot kopiert wurde, der ARN für den Quell-Cluster-Snapshot; andernfalls ein Nullwert.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Gibt den Status dieses Cluster-Snapshots an.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Gibt an, ob der Cluster-Snapshot verschlüsselt ist.

Typ: Boolesch

Erforderlich: Nein

StorageType

Speichertyp, der Ihrem Cluster-Snapshot zugeordnet ist

Informationen zu Speichertypen für Amazon DocumentDB-Cluster finden Sie unter Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Gültige Werte für den Speichertyp - `standard | iopt1`

Der Standardwert ist `standard`

Typ: Zeichenfolge

Erforderlich: Nein

VpcId

Stellt die Virtual Private Cloud (VPC) -ID bereit, die dem Cluster-Snapshot zugeordnet ist.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterSnapshotAttribute

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält den Namen und die Werte eines manuellen Cluster-Snapshot-Attributs.

Manuelle Cluster-Snapshot-Attribute werden verwendet, um andere AWS-Konten zur Wiederherstellung eines manuellen Cluster-Snapshots zu autorisieren.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AttributeName

Der Name des manuellen Cluster-Snapshot-Attributs.

Das angegebene Attribut `restore` bezieht sich auf die Liste der Personen AWS-Konten , die berechtigt sind, den manuellen Cluster-Snapshot zu kopieren oder wiederherzustellen.

Typ: Zeichenfolge

Erforderlich: Nein

AttributeValues.AttributeValue.N

Die Werte für das manuelle Cluster-Snapshot-Attribut.

Wenn das `AttributeName` Feld auf gesetzt ist`restore`, gibt dieses Element eine Liste IDs der Personen zurück AWS-Konten , die berechtigt sind, den manuellen Cluster-Snapshot zu kopieren oder wiederherzustellen. Wenn in der Liste der Wert von `all` steht, ist der manuelle Cluster-Snapshot öffentlich und kann von allen AWS-Konto kopiert oder wiederhergestellt werden.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBClusterSnapshotAttributesResult

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu den Attributen, die einem Cluster-Snapshot zugeordnet sind.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

DBClusterSnapshotAttributes.DBClusterSnapshotAttribute.N

Die Liste der Attribute und Werte für den Cluster-Snapshot.

Typ: Array von DBClusterSnapshotAttribute-Objekten

Erforderlich: Nein

DBClusterSnapshotIdentifier

Die ID des Cluster-Snapshots, für den die Attribute gelten.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBEngineVersion

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Engine-Version.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

DBEngineDescription

Die Beschreibung der Datenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

DBEngineVersionDescription

Die Beschreibung der Datenbank-Engine-Version.

Typ: Zeichenfolge

Erforderlich: Nein

DBParameterGroupFamily

Der Name der Parametergruppenfamilie für die Datenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Der Name der Datenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die Versionsnummer des Datenbank-Engines.

Typ: Zeichenfolge

Erforderlich: Nein

ExportableLogTypes.member.N

Die Protokolltypen, die die Datenbank-Engine für den Export nach Amazon CloudWatch Logs zur Verfügung stellt.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SupportedCACertificateIdentifiers.member.N

Eine Liste der unterstützten CA-Zertifikatsbezeichner.

Weitere Informationen finden Sie unter Aktualisieren Ihrer Amazon DocumentDB-TLS-Zertifikate und Verschlüsseln von Daten bei der Übertragung im Amazon DocumentDB-Entwicklerhandbuch.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SupportsCertificateRotationWithoutRestart

Gibt an, ob die Engine-Version das Rotieren des Serverzertifikats ohne Neustart der DB-Instance unterstützt.

Typ: Boolesch

Erforderlich: Nein

SupportsLogExportsToCloudwatchLogs

Ein Wert, der angibt, ob die Engine-Version das Exportieren der in `ExportableLogTypes` to CloudWatch Logs angegebenen Protokolltypen unterstützt.

Typ: Boolesch

Erforderlich: Nein

ValidUpgradeTarget.UpgradeTarget.N

Eine Liste der Engine-Versionen, auf die diese Datenbank-Engine-Version aktualisiert werden kann.

Typ: Array von UpgradeTarget-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBInstance

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Instanz.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

AutoMinorVersionUpgrade

Trifft nicht zu. Dieser Parameter gilt nicht für Amazon DocumentDB. Amazon DocumentDB führt unabhängig vom eingestellten Wert keine kleineren Versions-Upgrades durch.

Typ: Boolesch

Erforderlich: Nein

AvailabilityZone

Gibt den Namen der Availability Zone an, in der sich die Instance befindet.

Typ: Zeichenfolge

Erforderlich: Nein

BackupRetentionPeriod

Gibt die Anzahl der Tage an, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

CACertificateIdentifier

Die ID des Zertifizierungsstellenzertifikats für diese DB-Instance.

Typ: Zeichenfolge

Erforderlich: Nein

CertificateDetails

Die Details des Serverzertifikats der DB-Instance.

Typ: CertificateDetails Objekt

Erforderlich: Nein

CopyTagsToSnapshot

Ein Wert, der angibt, ob Tags aus der DB-Instance in Snapshots der DB-Instance kopiert werden sollen. Standardmäßig werden Tags nicht kopiert.

Typ: Boolesch

Erforderlich: Nein

DBClusterIdentifier

Enthält den Namen des Clusters, dem die Instance angehört, wenn die Instance Mitglied eines Clusters ist.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceArn

Der Amazon-Ressourcenname (ARN) für die Instance.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceClass

Enthält den Namen der Rechen- und Speicherkapazitätsklasse der Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceIdentifier

Enthält eine vom Benutzer bereitgestellte Datenbank-ID. Dieser Bezeichner ist der eindeutige Schlüssel, der eine Instanz identifiziert.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceStatus

Gibt den aktuellen Status dieser Datenbank an.

Typ: Zeichenfolge

Erforderlich: Nein

DbiResourceId

Der AWS-Region-eindeutige, unveränderliche Bezeichner für die Instanz. Dieser Bezeichner befindet AWS CloudTrail sich in Protokolleinträgen, wenn auf den AWS KMS Schlüssel für die Instanz zugegriffen wird.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroup

Gibt Informationen über die Subnetzgruppe an, die der Instance zugeordnet ist, einschließlich des Namens, der Beschreibung und der Subnetze in der Subnetzgruppe.

Typ: DBSubnetGroup Objekt

Erforderlich: Nein

EnabledCloudwatchLogsExports.member.N

Eine Liste der Protokolltypen, für deren Export diese Instanz in Logs konfiguriert ist. CloudWatch

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Endpoint

Gibt den Verbindungsendpunkt an.

Typ: Endpoint Objekt

Erforderlich: Nein

Engine

Gibt den Namen der Datenbank-Engine an, die für diese Instanz verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Erforderlich: Nein

InstanceCreateTime

Gibt Datum und Uhrzeit der Erstellung der Instanz an.

Typ: Zeitstempel

Erforderlich: Nein

KmsKeyId

Falls `StorageEncrypted` ja`true`, die AWS KMS Schlüssel-ID für die verschlüsselte Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

LatestRestorableTime

Gibt den letzten Zeitpunkt an, zu dem eine Datenbank mit point-in-time restore wiederhergestellt werden kann.

Typ: Zeitstempel

Erforderlich: Nein

PendingModifiedValues

Gibt an, dass Änderungen an der Instanz ausstehen. Dieses Element ist nur enthalten, wenn Änderungen ausstehen. Spezifische Änderungen werden von Unterelementen identifiziert.

Typ: PendingModifiedValues Objekt

Erforderlich: Nein

PerformanceInsightsEnabled

`true`Wird auf gesetzt, wenn Amazon RDS Performance Insights für die DB-Instance aktiviert ist, und andernfalls`false`.

Typ: Boolesch

Erforderlich: Nein

PerformanceInsightsKMSKeyId

Die AWS KMS Schlüssel-ID für die Verschlüsselung von Performance Insights Insights-Daten. Die AWS KMS Schlüssel-ID ist der Amazon-Ressourcenname (ARN), die AWS KMS Schlüssel-ID oder der AWS KMS Schlüsselalias für den AWS KMS Verschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredBackupWindow

Gibt den täglichen Zeitraum in koordinierter Weltzeit (UTC) an, in dem automatische Sicherungen erstellt werden, wenn automatische Sicherungen aktiviert sind, gemäß `BackupRetentionPeriod`.

Typ: Zeichenfolge

Erforderlich: Nein

PreferredMaintenanceWindow

Gibt den wöchentlichen Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time) an.

Typ: Zeichenfolge

Erforderlich: Nein

PromotionTier

Ein Wert, der die Reihenfolge angibt, in der ein Amazon DocumentDB DocumentDB-Replikat nach einem Ausfall der vorhandenen primären Instance zur primären Instance heraufgestuft wird.

Typ: Ganzzahl

Erforderlich: Nein

PubliclyAccessible

Nicht unterstützt Amazon DocumentDB unterstützt derzeit keine öffentlichen Endgeräte. Der Wert von `PubliclyAccessible` ist immer. `false`

Typ: Boolesch

Erforderlich: Nein

StatusInfos.DBInstanceStatusInfo.N

Der Status einer Read Replica. Wenn es sich bei der Instanz nicht um ein Read Replica handelt, ist dieses Feld leer.

Typ: Array von DBInstanceStatusInfo-Objekten

Erforderlich: Nein

StorageEncrypted

Gibt an, ob die Instanz verschlüsselt ist oder nicht.

Typ: Boolesch

Erforderlich: Nein

VpcSecurityGroups.VpcSecurityGroupMembership.N

Stellt eine Liste der VPC-Sicherheitsgruppenelemente bereit, zu denen die Instance gehört.

Typ: Array von VpcSecurityGroupMembership-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBInstanceStatusInfo

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt eine Liste mit Statusinformationen für eine Instanz bereit.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Message

Details des Fehlers, wenn ein Fehler bei der Instance auftritt. Wenn die Instance keinen Fehlerstatus aufweist, ist dieser Wert leer.

Typ: Zeichenfolge

Erforderlich: Nein

Normal

Ein boolescher Wert, der angibt, `true` ob die Instanz normal funktioniert oder `false` ob sich die Instanz in einem Fehlerstatus befindet.

Typ: Boolesch

Erforderlich: Nein

Status

Status der Instanz. Bei einem `StatusType` Read-Replikat können die Werte „Fehler“`replicating`, „oder`terminated`“ `stopped` lauten.

Typ: Zeichenfolge

Erforderlich: Nein

StatusType

Dieser Wert ist derzeit ""“`read replication`.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DBSubnetGroup

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Subnetzgruppe.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBSubnetGroupArn

Der Amazon-Ressourcenname (ARN) für die DB-Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupDescription

Stellt die Beschreibung der Subnetzgruppe bereit.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupName

Name der Subnetzgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

SubnetGroupStatus

Gibt den Status der Subnetzgruppe an.

Typ: Zeichenfolge

Erforderlich: Nein

Subnets.Subnet.N

Detaillierte Informationen zu einem oder mehreren Subnetzen innerhalb einer Subnetzgruppe.

Typ: Array von Subnet-Objekten

Erforderlich: Nein

VpcId

Stellt die Virtual Private Cloud (VPC) -ID der Subnetzgruppe bereit.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Endpoint

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Netzwerkinformationen für den Zugriff auf einen Cluster oder eine Instanz. Client-Programme müssen einen gültigen Endpunkt angeben, um auf diese Amazon DocumentDB DocumentDB-Ressourcen zuzugreifen.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Address

Gibt die DNS-Adresse der Instanz an.

Typ: Zeichenfolge

Erforderlich: Nein

HostedZoneId

Gibt die ID an, die Amazon Route 53 zuweist, wenn Sie eine gehostete Zone erstellen.

Typ: Zeichenfolge

Erforderlich: Nein

Port

Gibt die Portnummer an, die von der Datenbank-Engine überwacht wird.

Typ: Ganzzahl

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EngineDefaults

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Enthält das Ergebnis eines erfolgreichen Aufrufs der `DescribeEngineDefaultClusterParameters` Operation.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBParameterGroupFamily

Der Name der Cluster-Parametergruppenfamilie, für die die Engine-Parameterinformationen zurückgegeben werden sollen.

Typ: Zeichenfolge

Erforderlich: Nein

Marker

Ein optionales Paginierungstoken, das von einer vorherigen Anforderung bereitgestellt wird. Wenn Sie diesen Parameter angeben, enthält die Antwort nur die Datensätze zwischen der Markierung und dem durch `MaxRecords` angegebenen Wert.

Typ: Zeichenfolge

Erforderlich: Nein

Parameters.Parameter.N

Die Parameter einer bestimmten Cluster-Parametergruppenfamilie.

Typ: Array von Parameter-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Event

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Veranstaltung.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

Date

Legt das Datum und die Uhrzeit des Ereignisses fest.

Typ: Zeitstempel

Erforderlich: Nein

EventCategories.EventCategory.N

Legt die Kategorie für das Ereignis fest.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Message

Stellt den Text dieses Ereignisses bereit.

Typ: Zeichenfolge

Erforderlich: Nein

SourceArn

Der Amazon-Ressourcenname (ARN) für das Ereignis.

Typ: Zeichenfolge

Erforderlich: Nein

SourceIdentifier

Stellt die Kennung für die Quelle des Ereignisses bereit.

Typ: Zeichenfolge

Erforderlich: Nein

SourceType

Gibt den Quelltyp für dieses Ereignis an.

Typ: Zeichenfolge

Zulässige Werte: `db-instance | db-parameter-group | db-security-group | db-snapshot | db-cluster | db-cluster-snapshot`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EventCategoriesMap

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein Ereignisquellentyp, zusammen mit einem oder mehreren Namen für die Ereigniskategorie.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

EventCategories.EventCategory.N

Die Ereigniskategorien für den angegebenen Quelltyp.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SourceType

Der Quelltyp, zu dem die zurückgegebenen Kategorien gehören.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EventSubscription

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einer Veranstaltung, die Sie abonniert haben.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

CustomerAwsId

Das AWS Kundenkonto, das mit dem Amazon DocumentDB DocumentDB-Abonnement für Ereignisbenachrichtigungen verknüpft ist.

Typ: Zeichenfolge

Erforderlich: Nein

CustSubscriptionId

Die Abonnement-ID für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Erforderlich: Nein

Enabled

Ein boolescher Wert, der angibt, ob das Abonnement aktiviert ist. Der Wert von `true` gibt an, dass das Abonnement aktiviert ist.

Typ: Boolesch

Erforderlich: Nein

EventCategoriesList.EventCategory.N

Eine Liste der Ereigniskategorien für das Amazon DocumentDB DocumentDB-Abonnement für Ereignisbenachrichtigungen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

EventSubscriptionArn

Der Amazon-Ressourcenname (ARN) für das Ereignisabonnement.

Typ: Zeichenfolge

Erforderlich: Nein

SnsTopicArn

Das Thema ARN des Amazon DocumentDB DocumentDB-Abonnements für Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Erforderlich: Nein

SourceIdsList.SourceId.N

Eine Quellenliste IDs für das Amazon DocumentDB DocumentDB-Abonnement für Ereignisbenachrichtigungen.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

SourceType

Der Quelltyp für das Amazon DocumentDB DocumentDB-Abonnement für Ereignisbenachrichtigungen.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Der Status des Abonnements für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen.

Einschränkungen:

Kann einer der folgenden sein:`creating`,,`modifying`,`deleting`,`active`, `no-permission` `topic-not-exist`

Der `no-permission` Status gibt an, dass Amazon DocumentDB nicht mehr berechtigt ist, Beiträge zum SNS-Thema zu veröffentlichen. Der `topic-not-exist` Status gibt an, dass das Thema nach der Erstellung des Abonnements gelöscht wurde.

Typ: Zeichenfolge

Erforderlich: Nein

SubscriptionCreationTime

Der Zeitpunkt, zu dem das Abonnement für Amazon DocumentDB DocumentDB-Ereignisbenachrichtigungen erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Filter

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein benannter Satz von Filterwerten, der verwendet wird, um eine spezifischere Ergebnisliste zurückzugeben. Sie können einen Filter verwenden, um eine Gruppe von Ressourcen anhand bestimmter Kriterien zuzuordnen, z. IDs B.

Platzhalter werden in Filtern nicht unterstützt.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Name

Der Name des Filters Bei Filternamen wird zwischen Groß- und Kleinschreibung unterschieden.

Typ: Zeichenfolge

Erforderlich: Ja

Values.Value.N

Ein oder mehrere Filterwerte. Bei Filterwerten muss die Groß- und Kleinschreibung beachtet werden.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalCluster

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Ein Datentyp, der einen globalen Amazon DocumentDB-Cluster darstellt.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DatabaseName

Der Standarddatenbankname innerhalb des neuen globalen Clusters.

Typ: Zeichenfolge

Erforderlich: Nein

DeletionProtection

Die Löschschutzeinstellung für den neuen globalen Cluster.

Typ: Boolesch

Erforderlich: Nein

Engine

Die Amazon DocumentDB DocumentDB-Datenbank-Engine, die vom globalen Cluster verwendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Erforderlich: Nein

GlobalClusterArn

Der Amazon-Ressourcenname (ARN) für den globalen Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

GlobalClusterIdentifier

Enthält eine vom Benutzer angegebene globale Cluster-ID. Dieser Bezeichner ist der eindeutige Schlüssel, der einen globalen Cluster identifiziert.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 255 Zeichen.

Pattern: `[A-Za-z][0-9A-Za-z-:._]*`

Erforderlich: Nein

GlobalClusterMembers.GlobalClusterMember.N

Die Liste der Cluster IDs für sekundäre Cluster innerhalb des globalen Clusters. Derzeit auf einen Artikel beschränkt.

Typ: Array von GlobalClusterMember-Objekten

Erforderlich: Nein

GlobalClusterResourceId

Der AWS-Region regionsspezifische, unveränderliche Bezeichner für den globalen Datenbankcluster. Diese Kennung befindet AWS CloudTrail sich in Protokolleinträgen, wenn auf den AWS KMS Kundenhauptschlüssel (CMK) für den Cluster zugegriffen wird.

Typ: Zeichenfolge

Erforderlich: Nein

Status

Gibt den aktuellen Status dieses globalen Clusters an.

Typ: Zeichenfolge

Erforderlich: Nein

StorageEncrypted

Die Speicherverschlüsselungseinstellung für den globalen Cluster.

Typ: Boolesch

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalClusterMember

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Eine Datenstruktur mit Informationen über alle primären und sekundären Cluster, die einem globalen Amazon DocumentDB-Cluster zugeordnet sind.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

DBClusterArn

Der Amazon-Ressourcenname (ARN) für jeden Amazon DocumentDB-Cluster.

Typ: Zeichenfolge

Erforderlich: Nein

IsWriter

Gibt an, ob der Amazon DocumentDB-Cluster der primäre Cluster ist (d. h. über Lese- und Schreibfähigkeit verfügt) für den globalen Amazon DocumentDB-Cluster, mit dem er verknüpft ist.

Typ: Boolesch

Erforderlich: Nein

Readers.member.N

Der Amazon-Ressourcenname (ARN) für jeden schreibgeschützten sekundären Cluster, der dem globalen Amazon DocumentDB-Cluster zugeordnet ist.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OrderableDBInstanceOption

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Die Optionen, die für eine Instanz verfügbar sind.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AvailabilityZones.AvailabilityZone.N

Eine Liste der Availability Zones für eine Instance.

Typ: Array von AvailabilityZone-Objekten

Erforderlich: Nein

DBInstanceClass

Die Instanzklasse für eine Instance.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Der Engine-Typ einer Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die Engine-Version einer Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

LicenseModel

Das Lizenzmodell für eine Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

StorageType

Der Speichertyp, der dem DB-Cluster zugeordnet werden soll

Typ: Zeichenfolge

Erforderlich: Nein

Vpc

Gibt an, ob sich eine Instanz in einer Virtual Private Cloud (VPC) befindet.

Typ: Boolesch

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Parameter

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem einzelnen Parameter.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

AllowedValues

Gibt den gültigen Wertebereich für den Parameter an.

Typ: Zeichenfolge

Erforderlich: Nein

ApplyMethod

Gibt an, wann Parameteraktualisierungen angewendet werden können.

Typ: Zeichenfolge

Zulässige Werte: `immediate | pending-reboot`

Erforderlich: Nein

ApplyType

Gibt den Typ der motorspezifischen Parameter an.

Typ: Zeichenfolge

Erforderlich: Nein

DataType

Gibt den gültigen Datentyp für den Parameter an.

Typ: Zeichenfolge

Erforderlich: Nein

Description

Stellt eine Beschreibung des Parameters bereit.

Typ: Zeichenfolge

Erforderlich: Nein

IsModifiable

Gibt an, ob der Parameter geändert werden kann oder nicht (`true` oder `false`). Einige Parameter wirken sich auf die Sicherheit oder die betrieblichen Abläufe aus und können nicht geändert werden.

Typ: Boolesch

Erforderlich: Nein

MinimumEngineVersion

Die älteste Engine-Version, auf die der Parameter angewendet werden kann.

Typ: Zeichenfolge

Erforderlich: Nein

ParameterName

Gibt den Namen des Parameters an.

Typ: Zeichenfolge

Erforderlich: Nein

ParameterValue

Gibt den Wert des Parameters an.

Typ: Zeichenfolge

Erforderlich: Nein

Source

Gibt die Quelle des Parameterwerts an.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im Folgenden AWS SDKs:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingCloudwatchLogsExports

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Eine Liste der Protokolltypen, deren Konfiguration noch aussteht. Diese Protokolltypen werden gerade aktiviert oder deaktiviert.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

LogTypesToDisable.member.N

Protokolltypen, die gerade aktiviert werden. Nachdem sie aktiviert wurden, werden diese Protokolltypen nach Amazon CloudWatch Logs exportiert.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

LogTypesToEnable.member.N

Protokolltypen, die gerade deaktiviert werden. Nachdem sie deaktiviert wurden, werden diese Protokolltypen nicht nach CloudWatch Logs exportiert.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingMaintenanceAction

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt Informationen über eine ausstehende Wartungsaktion für eine Ressource bereit.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Action

Der Typ der ausstehenden Wartungsaktion, die für die Ressource verfügbar ist.

Typ: Zeichenfolge

Erforderlich: Nein

AutoAppliedAfterDate

Das Datum des Wartungsfensters, in dem die Aktion angewendet wird. Die Wartungsaktion wird während ihres ersten Wartungsfensters nach diesem Datum auf die Ressource angewendet. Wenn dieses Datum angegeben ist, werden alle `next-maintenance`-Opt-In-Anfragen ignoriert.

Typ: Zeitstempel

Erforderlich: Nein

CurrentApplyDate

Das Datum des Inkrafttretens, wenn die ausstehende Wartungsaktion auf die Ressource angewendet wird.

Typ: Zeitstempel

Erforderlich: Nein

Description

Eine Beschreibung, die weitere Details zu der Wartungsaktion bereitstellt.

Typ: Zeichenfolge

Erforderlich: Nein

ForcedApplyDate

Das Datum, an dem die Wartungsaktion automatisch angewendet wird. Die Wartungsaktion wird ungeachtet des Wartungsfensters für die Ressource an diesem Datum auf die Ressource angewendet. Wenn dieses Datum angegeben ist, werden alle `immediate`-Opt-In-Anfragen ignoriert.

Typ: Zeitstempel

Erforderlich: Nein

OptInStatus

Gibt den Typ der Opt-in-Anforderung an, die für die Ressource empfangen wurde.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingModifiedValues

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Eine oder mehrere geänderte Einstellungen für eine Instanz. Diese geänderten Einstellungen wurden angefordert, aber noch nicht angewendet.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AllocatedStorage

Enthält die neue `AllocatedStorage` Größe für die Instanz, die angewendet wird oder gerade angewendet wird.

Typ: Ganzzahl

Erforderlich: Nein

BackupRetentionPeriod

Gibt die ausstehende Anzahl von Tagen an, die automatische Sicherungen aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

CACertificateIdentifier

Gibt den Bezeichner des Zertifizierungsstellenzertifikats (CA) für die DB-Instance an.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceClass

Enthält das neue `DBInstanceClass` für die Instance, das angewendet wird oder gerade angewendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

DBInstanceIdentifier

Enthält die neuen `DBInstanceIdentifier` Informationen für die Instanz, die angewendet wird oder gerade angewendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

DBSubnetGroupName

Die neue Subnetzgruppe für die Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Gibt die Version der Datenbank-Engine an.

Typ: Zeichenfolge

Erforderlich: Nein

Iops

Gibt den neuen Wert für bereitgestellte IOPS für die Instanz an, die angewendet wird oder gerade angewendet wird.

Typ: Ganzzahl

Erforderlich: Nein

LicenseModel

Das Lizenzmodell für die Instanz.

Zulässige Werte: `license-included`, `bring-your-own-license`, `general-public-license`

Typ: Zeichenfolge

Erforderlich: Nein

MasterUserPassword

Enthält die ausstehende oder derzeit in Bearbeitung befindliche Änderung der Master-Anmeldeinformationen für die Instanz.

Typ: Zeichenfolge

Erforderlich: Nein

MultiAZ

Zeigt an, dass die Single-AZ-Instance zu einer Multi-AZ-Bereitstellung wechseln soll.

Typ: Boolesch

Erforderlich: Nein

PendingCloudwatchLogsExports

Eine Liste der Protokolltypen, deren Konfiguration noch aussteht. Diese Protokolltypen werden gerade aktiviert oder deaktiviert.

Typ: PendingCloudwatchLogsExports Objekt

Erforderlich: Nein

Port

Gibt den ausstehenden Port für die Instanz an.

Typ: Ganzzahl

Erforderlich: Nein

StorageType

Gibt den Speichertyp an, der der Instanz zugeordnet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ResourcePendingMaintenanceActions

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Stellt die Ausgabe von darApplyPendingMaintenanceAction.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

PendingMaintenanceActionDetails.PendingMaintenanceAction.N

Eine Liste mit Details zu den ausstehenden Wartungsaktionen für die Ressource.

Typ: Array von PendingMaintenanceAction-Objekten

Erforderlich: Nein

ResourceIdentifier

Der Amazon-Ressourcenname (ARN) der Ressource, für die Wartungsaktionen ausstehen.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Subnet

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Detaillierte Informationen zu einem Subnetz.

Inhalt

> **Note**
>
> In der folgenden Liste werden zunächst die erforderlichen Parameter beschrieben.

SubnetAvailabilityZone

Gibt die Availability Zone für das Subnetz an.

Typ: AvailabilityZone Objekt

Erforderlich: Nein

SubnetIdentifier

Gibt die Kennung des Subnetzes an.

Typ: Zeichenfolge

Erforderlich: Nein

SubnetStatus

Gibt den Status des Subnetzes an.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

Tag

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Einer Amazon DocumentDB DocumentDB-Ressource zugewiesene Metadaten, die aus einem Schlüssel-Wert-Paar bestehen.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Key

Der erforderliche Name des Tags. Der Zeichenkettenwert kann eine Länge von 1 bis 128 Unicode-Zeichen haben und darf nicht mit "" oder `aws:` "`rds:`“ vorangestellt werden. Die Zeichenfolge kann nur den Satz von Unicode-Buchstaben, Ziffern, Leerzeichen, '_', 'enthalten . ','/',' = ',' + ',' - '(Java-Regex: „^ ([\\ p {L}\\ p {Z}\\ p {N} _.: /=+\\ -] *) $“).

Typ: Zeichenfolge

Erforderlich: Nein

Value

Der optionale Wert des Tags. Der Zeichenkettenwert kann eine Länge von 1 bis 256 Unicode-Zeichen haben und ihm darf kein Präfix "" oder `aws:` "`rds:`“ vorangestellt werden. Die Zeichenfolge kann nur den Satz von Unicode-Buchstaben, Ziffern, Leerzeichen, '_', 'enthalten . ','/',' = ',' + ',' - '(Java-Regex: „^ ([\\ p {L}\\ p {Z}\\ p {N} _.: /=+\\ -] *) $“).

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpgradeTarget

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Die Version der Datenbank-Engine, auf die eine Instanz aktualisiert werden kann.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

AutoUpgrade

Ein Wert, der angibt, ob die Zielversion auf Quell-DB-Instances angewendet wird, für die die `AutoMinorVersionUpgrade` Einstellung auf festgelegt wurde`true`.

Typ: Boolesch

Erforderlich: Nein

Description

Die Version der Datenbank-Engine, auf die eine Instance aktualisiert werden kann.

Typ: Zeichenfolge

Erforderlich: Nein

Engine

Der Name der Upgrade-Zieldatenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

EngineVersion

Die Versionsnummer der Upgrade-Zieldatenbank-Engine.

Typ: Zeichenfolge

Erforderlich: Nein

IsMajorVersionUpgrade

Ein Wert, der angibt, ob eine Datenbank-Engine auf eine Hauptversion aktualisiert wird.

Typ: Boolesch

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

VpcSecurityGroupMembership

Bedienung: Amazon DocumentDB (with MongoDB compatibility)

Wird als Antwortelement für Abfragen zur Mitgliedschaft in einer Virtual Private Cloud (VPC) - Sicherheitsgruppe verwendet.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

Status

Der Status der VPC-Sicherheitsgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

VpcSecurityGroupId

Der Name der VPC-Sicherheitsgruppe.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon DocumentDB Elastic Clusters

Die folgenden Datentypen werden von Amazon DocumentDB Elastic Clusters unterstützt:

- Cluster
- ClusterInList
- ClusterSnapshot
- ClusterSnapshotInList
- PendingMaintenanceActionDetails
- ResourcePendingMaintenanceAction
- Shard
- ValidationExceptionField

Cluster

Bedienung: Amazon DocumentDB Elastic Clusters

Gibt Informationen über einen bestimmten elastischen Cluster zurück.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

adminUserName

Der Name des Elastic Cluster-Administrators.

Typ: Zeichenfolge

Erforderlich: Ja

authType

Der Authentifizierungstyp für den Elastic Cluster.

Typ: Zeichenfolge

Zulässige Werte: `PLAIN_TEXT | SECRET_ARN`

Erforderlich: Ja

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

clusterEndpoint

Die URL, die für die Verbindung mit dem Elastic Cluster verwendet wird.

Typ: Zeichenfolge

Erforderlich: Ja

clusterName

Der Name des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

createTime

Der Zeitpunkt, zu dem der elastische Cluster in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

kmsKeyId

Die KMS-Schlüssel-ID, die zum Verschlüsseln des Elastic Clusters verwendet werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

preferredMaintenanceWindow

Der wöchentliche Zeitraum, in dem Systemwartungen durchgeführt werden können, in UTC (Universal Coordinated Time).

Format: `ddd:hh24:mi-ddd:hh24:mi`

Typ: Zeichenfolge

Erforderlich: Ja

shardCapacity

Die Anzahl der V, die jedem Elastic Cluster-Shard CPUs zugewiesen sind. Das Maximum ist 64. Zulässige Werte sind 2, 4, 8, 16, 32, 64.

Typ: Ganzzahl

Erforderlich: Ja

shardCount

Die Anzahl der Shards, die dem elastischen Cluster zugewiesen sind. Das Maximum ist 32.

Typ: Ganzzahl

Erforderlich: Ja

status

Der Status des elastischen Clusters.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED | MAINTENANCE | INACCESSIBLE_ENCRYPTION_CREDENTIALS_RECOVERABLE`

Erforderlich: Ja

subnetIds

Das EC2 Amazon-Subnetz IDs für den Elastic Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

vpcSecurityGroupIds

Eine Liste der EC2 VPC-Sicherheitsgruppen, die diesem elastischen Cluster zugeordnet sind.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

backupRetentionPeriod

Die Anzahl der Tage, für die automatische Snapshots aufbewahrt werden.

Typ: Ganzzahl

Erforderlich: Nein

preferredBackupWindow

Der tägliche Zeitraum, in dem automatische Backups erstellt werden, wenn automatische Backups aktiviert sind, wie von `backupRetentionPeriod` festgelegt.

Typ: Zeichenfolge

Erforderlich: Nein

shardInstanceCount

Die Anzahl der Replikatinstanzen, die für alle Shards im Cluster gelten. Ein `shardInstanceCount` Wert von 1 bedeutet, dass es eine Writer-Instanz gibt, und alle weiteren Instanzen sind Replikate, die für Lesevorgänge und zur Verbesserung der Verfügbarkeit verwendet werden können.

Typ: Ganzzahl

Erforderlich: Nein

shards

Die Gesamtzahl der Shards im Cluster.

Typ: Array von Shard-Objekten

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterInList

Bedienung: Amazon DocumentDB Elastic Clusters

Eine Liste elastischer Amazon DocumentDB-Cluster.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

clusterArn

Der ARN-Bezeichner des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

clusterName

Der Name des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der Status des elastischen Clusters.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED | MAINTENANCE | INACCESSIBLE_ENCRYPTION_CREDENTIALS_RECOVERABLE`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterSnapshot

Bedienung: Amazon DocumentDB Elastic Clusters

Gibt Informationen zu einem bestimmten Elastic Cluster-Snapshot zurück.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

adminUserName

Der Name des Elastic Cluster-Administrators.

Typ: Zeichenfolge

Erforderlich: Ja

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

clusterCreationTime

Der Zeitpunkt, zu dem der elastische Cluster in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

kmsKeyId

Die Kennung für den KMS-Schlüssel ist der Amazon-Ressourcenname (ARN) für den KMS-Verschlüsselungsschlüssel. Wenn Sie einen Cluster mit demselben Amazon-Konto erstellen, dem dieser KMS-Verschlüsselungsschlüssel gehört, können Sie den KMS-Schlüsselalias anstelle des ARN als KMS-Verschlüsselungsschlüssel verwenden. Wenn hier kein Verschlüsselungsschlüssel angegeben ist, verwendet Amazon DocumentDB den Standardverschlüsselungsschlüssel,

den KMS für Ihr Konto erstellt. Ihr Konto hat für jede Amazon-Region einen anderen Standard-Verschlüsselungsschlüssel.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotArn

Die ARN-ID des Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotCreationTime

Der Zeitpunkt, zu dem der Elastic Cluster-Snapshot in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotName

Der Name des Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der Status des Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED | MAINTENANCE | INACCESSIBLE_ENCRYPTION_CREDENTIALS_RECOVERABLE`

Erforderlich: Ja

subnetIds

Das EC2 Amazon-Subnetz IDs für den Elastic Cluster.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

vpcSecurityGroupIds

Eine Liste von EC2 VPC-Sicherheitsgruppen, die dem Elastic Cluster zugeordnet werden sollen.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

snapshotType

Der Typ der Cluster-Snapshots, die zurückgegeben werden sollen. Sie können einen der folgenden Werte angeben:

- `automated`- Gibt alle Cluster-Snapshots zurück, die Amazon DocumentDB automatisch für Ihr AWS Konto erstellt hat.
- `manual`- Gibt alle Cluster-Snapshots zurück, die Sie manuell für Ihr Konto erstellt haben. AWS

Typ: Zeichenfolge

Zulässige Werte: `MANUAL | AUTOMATED`

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterSnapshotInList

Bedienung: Amazon DocumentDB Elastic Clusters

Eine Liste von Elastic Cluster-Snapshots.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

clusterArn

Die ARN-ID des elastischen Clusters.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotArn

Die ARN-ID des Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotCreationTime

Der Zeitpunkt, zu dem der Elastic Cluster-Snapshot in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

snapshotName

Der Name des Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der Status des Elastic Cluster-Snapshots.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED | MAINTENANCE | INACCESSIBLE_ENCRYPTION_CREDENTIALS_RECOVERABLE`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PendingMaintenanceActionDetails

Bedienung: Amazon DocumentDB Elastic Clusters

Ruft die Details der ausstehenden Wartungsaktionen ab.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

action

Zeigt die spezifische Aktion einer ausstehenden Wartungsaktion an.

Typ: Zeichenfolge

Erforderlich: Ja

autoAppliedAfterDate

Zeigt das Datum des Wartungsfensters an, an dem die Aktion angewendet wird. Die Wartungsaktion wird während ihres ersten Wartungsfensters nach diesem Datum auf die Ressource angewendet. Wenn dieses Datum angegeben ist, werden alle `NEXT_MAINTENANCE` `optInType` Anfragen ignoriert.

Typ: Zeichenfolge

Erforderlich: Nein

currentApplyDate

Zeigt das Datum an, an dem die ausstehende Wartungsaktion auf die Ressource angewendet wird.

Typ: Zeichenfolge

Erforderlich: Nein

description

Zeigt eine Beschreibung mit weiteren Einzelheiten zur Wartungsaktion an.

Typ: Zeichenfolge

Erforderlich: Nein

forcedApplyDate

Zeigt das Datum an, an dem die Wartungsaktion automatisch angewendet wird. Die Wartungsaktion wird ungeachtet des Wartungsfensters für die Ressource an diesem Datum auf die Ressource angewendet. Wenn dieses Datum angegeben ist, werden alle `IMMEDIATE` `optInType` Anfragen ignoriert.

Typ: Zeichenfolge

Erforderlich: Nein

optInStatus

Zeigt den Typ der `optInType` Anfrage an, die für die Ressource eingegangen ist.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ResourcePendingMaintenanceAction

Bedienung: Amazon DocumentDB Elastic Clusters

Stellt Informationen über eine ausstehende Wartungsaktion für eine Ressource bereit.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

pendingMaintenanceActionDetails

Stellt Informationen über eine ausstehende Wartungsaktion für eine Ressource bereit.

Typ: Array von PendingMaintenanceActionDetails-Objekten

Erforderlich: Nein

resourceArn

Der Amazon DocumentDB Amazon Resource Name (ARN) der Ressource, für die die ausstehende Wartungsaktion gilt.

Typ: Zeichenfolge

Erforderlich: Nein

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Shard

Bedienung: Amazon DocumentDB Elastic Clusters

Der Name der Scherbe.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

createTime

Die Zeit, zu der der Shard in UTC (Universal Coordinated Time) erstellt wurde.

Typ: Zeichenfolge

Erforderlich: Ja

shardId

Die ID des Shards.

Typ: Zeichenfolge

Erforderlich: Ja

status

Der aktuelle Status des Shards.

Typ: Zeichenfolge

Zulässige Werte: `CREATING | ACTIVE | DELETING | UPDATING | VPC_ENDPOINT_LIMIT_EXCEEDED | IP_ADDRESS_LIMIT_EXCEEDED | INVALID_SECURITY_GROUP_ID | INVALID_SUBNET_ID | INACCESSIBLE_ENCRYPTION_CREDS | INACCESSIBLE_SECRET_ARN | INACCESSIBLE_VPC_ENDPOINT | INCOMPATIBLE_NETWORK | MERGING | MODIFYING | SPLITTING | COPYING | STARTING | STOPPING | STOPPED | MAINTENANCE | INACCESSIBLE_ENCRYPTION_CREDENTIALS_RECOVERABLE`

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen finden Sie im AWS SDKs Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ValidationExceptionField

Bedienung: Amazon DocumentDB Elastic Clusters

Ein bestimmtes Feld, in dem eine bestimmte Validierungsausnahme aufgetreten ist.

Inhalt

> **Note**
>
> In der folgenden Liste werden zuerst die erforderlichen Parameter beschrieben.

message

Eine Fehlermeldung, die die Validierungsausnahme in diesem Feld beschreibt.

Typ: Zeichenfolge

Erforderlich: Ja

name

Der Name des Felds, in dem die Validierungsausnahme aufgetreten ist.

Typ: Zeichenfolge

Erforderlich: Ja

Weitere Informationen finden Sie unter:

Weitere Informationen zur Verwendung dieser API in einer der sprachspezifischen Sprachen AWS SDKs finden Sie im Folgenden:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Häufige Fehler

In diesem Abschnitt werden die Fehler aufgeführt, die bei den API-Aktionen aller AWS Dienste häufig auftreten. Informationen zu Fehlern, die spezifisch für eine API-Aktion für diesen Service sind, finden Sie unter dem Thema für diese API-Aktion.

AccessDeniedException

Sie haben keinen ausreichenden Zugriff zum Durchführen dieser Aktion.

HTTP Status Code: 400

IncompleteSignature

Die Anforderungssignatur entspricht nicht den AWS Standards.

HTTP Status Code: 400

InternalFailure

Die Anforderungsverarbeitung ist fehlgeschlagen, da ein unbekannter Fehler, eine Ausnahme oder ein Fehler aufgetreten ist.

HTTP Status Code: 500

InvalidAction

Die angeforderte Aktion oder Operation ist ungültig. Überprüfen Sie, ob die Aktion ordnungsgemäß eingegeben wurde.

HTTP Status Code: 400

InvalidClientTokenId

Das angegebene X.509-Zertifikat oder die angegebene AWS Zugriffsschlüssel-ID ist in unseren Aufzeichnungen nicht vorhanden.

HTTP Status Code: 403

NotAuthorized

Sie haben keine Berechtigung zum Ausführen dieser Aktion.

HTTP Status Code: 400

OptInRequired

Für die AWS Zugriffsschlüssel-ID ist ein Abonnement für den Dienst erforderlich.

HTTP Status Code: 403

RequestExpired

Die Anfrage erreichte den Service mehr als 15 Minuten nach dem Datumsstempel auf der Anfrage oder mehr als 15 Minuten nach dem Ablaufdatum der Anfrage (z. B. bei vorsignierter Anfrage URLs), oder der Datumsstempel auf der Anfrage liegt mehr als 15 Minuten in der future.

HTTP Status Code: 400

ServiceUnavailable

Die Anforderung ist aufgrund eines temporären Fehlers des Servers fehlgeschlagen.

HTTP Status Code: 503

ThrottlingException

Die Anforderung wurde aufgrund der Drosselung von Anforderungen abgelehnt.

HTTP Status Code: 400

ValidationError

Die Eingabe erfüllt nicht die von einem AWS Dienst angegebenen Einschränkungen.

HTTP Status Code: 400

Geläufige Parameter

Die folgende Liste enthält die Parameter, die alle Aktionen zum Signieren von Signature-Version-4-Anforderungen mit einer Abfragezeichenfolge verwenden. Alle aktionsspezifischen Parameter werden im Thema für diese Aktion aufgelistet. Weitere Informationen zu Signature Version 4 finden Sie unter Signieren von AWS API-Anfragen im IAM-Benutzerhandbuch.

Action

Die auszuführende Aktion.

Typ: Zeichenfolge

Erforderlich: Ja

Version

Die API-Version, für die die Anfrage geschrieben wurde, ausgedrückt im Format YYYY-MM-DD.

Typ: Zeichenfolge

Erforderlich: Ja

X-Amz-Algorithm

Der Hashalgorithmus, den Sie zum Erstellen der Anforderungssignatur verwendet haben.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Zulässige Werte: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

Der Wert des Anmeldeinformationsumfangs. Dabei handelt es sich um eine Zeichenfolge, die Ihren Zugriffsschlüssel, das Datum, die gewünschte Region und eine Zeichenfolge zur Beendigung („aws4_request“) beinhaltet. Der Wert wird im folgenden Format ausgedrückt: Zugriffsschlüssel/JJJJMMTT/Region/Service/aws4_request.

Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erstellen einer signierten AWS API-Anfrage.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Date

Das Datum, das zum Erstellen der Signatur verwendet wird. Das Format muss das ISO 8601-Basisformat (JJJJMMTT'T'SSMMSS'Z') sein. Beispielsweise ist die folgende Datums- und Uhrzeitangabe ein gültiger X-Amz-Date Wert:`20120325T120000Z`.

Bedingung: X-Amz-Date ist bei allen Anforderungen optional. Damit kann das Datum überschrieben werden, das zum Signieren von Anforderungen verwendet wird. Wenn der Date-Header im ISO 8601-Grundformat angegeben ist, X-Amz-Date ist dies nicht erforderlich. Wenn verwendet X-Amz-Date wird, überschreibt er immer den Wert des Date-Headers. Weitere Informationen finden Sie unter Elemente einer AWS API-Anforderungssignatur im IAM-Benutzerhandbuch.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Security-Token

Das temporäre Sicherheitstoken, das durch einen Aufruf von AWS Security Token Service (AWS STS) abgerufen wurde. Eine Liste der Services, die temporäre Sicherheits-Anmeldeinformationen von AWS STS unterstützen, finden Sie im IAM-Benutzerhandbuch unter AWS-Services , die mit IAM funktionieren.

Bedingung: Wenn Sie temporäre Sicherheitsanmeldedaten von verwenden AWS STS, müssen Sie das Sicherheitstoken angeben.

Typ: Zeichenfolge

Required: Conditional

X-Amz-Signature

Gibt die hex-codierte Signatur an, die aus der zu signierenden Zeichenfolge und dem abgeleiteten Signaturschlüssel berechnet wurde.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

X-Amz-SignedHeaders

Gibt alle HTTP-Header an, die als Teil der kanonischen Anforderung enthalten waren. Weitere Informationen zur Angabe signierter Header finden Sie unter Erstellen einer signierten AWS API-Anfrage im IAM-Benutzerhandbuch.

Bedingung: Geben Sie diesen Parameter an, wenn Sie Authentifizierungsinformationen in eine Abfragezeichenfolge anstatt in den HTTP-Autorisierungsheader aufnehmen.

Typ: Zeichenfolge

Required: Conditional

Versionshinweise

In diesen Versionshinweisen werden die Funktionen, Verbesserungen und Fehlerkorrekturen von Amazon DocumentDB nach Veröffentlichungsdatum beschrieben. Die Versionshinweise enthalten Updates für alle Versionen der Amazon DocumentDB DocumentDB-Engine, sobald sie verfügbar sind.

Sie können die aktuelle Patch-Version der Amazon DocumentDB DocumentDB-Engine ermitteln, indem Sie den folgenden Befehl ausführen:

```
db.runCommand({getEngineVersion: 1})
```

Wenn Ihr Cluster nicht auf der neuesten Version der Engine läuft, ist es wahrscheinlich, dass noch Wartungsarbeiten verfügbar sind, mit denen Ihre Engine aktualisiert werden soll. Weitere Informationen finden Sie Wartung von Amazon DocumentDB im Entwicklerhandbuch.

Sie können neue Amazon DocumentDB DocumentDB-Funktionen unter Was ist neu in der Datenbank? filtern. Seite. Wählen Sie für Produkte Amazon DocumentDB. Suchen Sie dann mit Schlüsselwörtern wie **`elastic clusters`** oder **`vector search`**.

Themen

- 18. Juni 2025
- 8. Mai 2025
- 2. April 2025
- 24. März 2025
- 6. Februar 2025
- 28. Januar 2025
- 15. Januar 2025
- 18. Dezember 2024
- 12. November 2024
- 6. November 2024
- 1. November 2024
- 22. Oktober 2024

- 18. September 2024
- 17. September 2024
- 22. August 2024
- 20. August 2024
- 8. August 2024
- 23. Juli 2024
- 22. Juli 2024
- 9. Juli 2024
- 8. Juli 2024
- 25. Juni 2024
- 29. Mai 2024
- 3. April 2024
- 22. Februar 2024
- 30. Januar 2024
- 10. Januar 2024
- 20. Dezember 2023
- 13. Dezember 2023
- 29. November 2023
- 21. November 2023
- 17. November 2023
- 6. November 2023
- 25. September 2023
- 20. September 2023
- 15. September 2023
- 11. September 2023
- 3. August 2023
- 13. Juli 2023
- 7. Juni 2023
- 10. Mai 2023

- 4. April 2023
- 22. März 2023
- 1. März 2023
- 27. Februar 2023
- 2. Februar 2023
- 30. November 2022
- 09. August 2022
- 25. Juli 2022
- 27. Juni 2022
- 29. April 2022
- 7. April 2022
- 16. März 2022
- 8. Februar 2022
- 24. Januar 2022
- 21. Januar 2022
- 25. Oktober 2021
- 24. Juni 2021
- 4. Mai 2021
- 15. Januar 2021
- 9. November 2020
- 30. Oktober 2020
- 22. September 2020
- 10. Juli 2020
- 30. Juni 2020

18. Juni 2025

Neues Feature

Amazon DocumentDB ist jetzt in der Region Israel (Tel Aviv) verfügbar.

8. Mai 2025

Neues Feature

Amazon DocumentDB ist jetzt in der Region Europa (Stockholm) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

2. April 2025

> **Note**
>
> Der folgende Amazon DocumentDB DocumentDB-Engine-Patch wird in den nächsten Wochen an alle Amazon DocumentDB DocumentDB-Regionen ausgeliefert. Wenn dieser Engine-Patch in Ihrer Region verfügbar ist, erhalten Sie eine Service-Patch-Benachrichtigung über das AWS Health Dashboard (AHD) im AWS Management Console und per E-Mail an die Root-Benutzer-E-Mail-Adresse Ihres AWS Kontos.
> Dieser Engine-Patch enthält die folgenden neuen Funktionen und Bugfixes. Bitte beachten Sie, dass die folgende Liste zusammen mit der entsprechenden unterstützenden Dokumentation aktualisiert werden kann, um zusätzliche Funktionsankündigungen aufzunehmen, sobald der Engine-Patch in allen Regionen verfügbar ist.

Neues Feature

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.14228)

Unterstützung für ungenutzte Speichergröße und Prozentsatz im Diagnosebefehl wurde hinzugefügt. `collStats`

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.11357)

Unterstützung für ungenutzte Speichergröße und Prozentsatz im Diagnosebefehl wurde hinzugefügt. `collStats`

Fehlerkorrekturen und andere Änderungen

Ein Fehler bei der Erstellung von Vektorindizes mit Hintergrundindizes wurde behoben.

24. März 2025

Neues Feature

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.11800)

Unterstützung für `postBatchResumeToken` In-Change-Streams hinzugefügt.

Weitere Informationen finden Sie unter Wiederaufnahme eines Change-Streams mit `postBatchResumeToken`.

6. Februar 2025

Neues Feature

Amazon DocumentDB ist jetzt in integriert AWS Toolkit for Visual Studio Code. Weitere Informationen finden Sie in diesem Blogbeitrag „Was ist neu“ und im AWS Toolkit for Visual Studio Code Benutzerhandbuch unter Arbeiten mit Amazon DocumentDB im Toolkit.

28. Januar 2025

Neues Feature

Für Konnektivität mit einem Klick ist Amazon DocumentDB jetzt AWS CloudShell für instanzbasierte und elastische Cluster integriert.

Weitere Informationen finden Sie in einer oder allen der folgenden Seiten:

- Erste Schritte mit Amazon DocumentDB
- Erste Schritte mit Amazon DocumentDB Elastic Clusters
- Was ist ein neuer Blogbeitrag
- Blogbeitrag mit technischen Anleitungen

15. Januar 2025

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.11800)

Amazon DocumentDB behält jetzt die Leseverfügbarkeit auch bei Neustarts der Writer-Instance bei. Reader-Instances bearbeiten nun weiterhin Leseanfragen bei Neustarts der Writer-Instance.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.11800)

- Der `killOp` Befehl zur Behandlung eines Sonderfalls von Masseneinfügungen wurde korrigiert.
- Verbesserte Speichernutzung im Zusammenhang mit Netzwerk-E/A auf Amazon DocumentDB DocumentDB-Instances.
- Der `count` Befehl für `$text` Filterabfragen wurde behoben.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.11096)

- Der `killOp` Befehl zur Behandlung eines Sonderfalls von Masseneinfügungen wurde korrigiert.
- Verbesserte Speichernutzung im Zusammenhang mit Netzwerk-E/A auf Amazon DocumentDB DocumentDB-Instances.

18. Dezember 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.5721)

Instanzbasierte Amazon DocumentDB DocumentDB-Cluster unterstützen jetzt R6GD-Instances mit NVMe -Unterstützung.

Weitere Informationen finden Sie unter NVMe-gestützte Instances und in diesem Blogbeitrag „Was ist neu“.

12. November 2024

Neue Features

Elastische Amazon DocumentDB-Cluster unterstützen jetzt Hintergrundindizes.

6. November 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.11051)

- Unterstützung für minimale TLS-Versionen `tls1.2+` und `tls1.3+` Cluster-Parameter hinzugefügt. `tls`
- Unterstützung für die Zeichen Dollar ($) und Punkt (.) in Feldnamen aktiviert. Informationen zu funktionalen Unterschieden finden Sie unterDollar ($) und Punkt (.) in Feldnamen.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.10980)

- Unterstützung für minimale TLS-Versionen `tls1.2+` und `tls1.3+` Cluster-Parameter hinzugefügt. `tls`

1. November 2024

Neues Feature

Amazon DocumentDB unterstützt jetzt Wartungsaktionen für elastische Cluster. Weitere Informationen finden Sie unter Wartung elastischer Amazon DocumentDB-Cluster.

22. Oktober 2024

Neues Feature

Amazon DocumentDB unterstützt jetzt Durchsatzmetriken für `StorageNetworkReceiveThroughput` SpeichernetzwerkeStorageNetworkTransmitThroughput, und`StorageNetworkThroughput`. Weitere Informationen finden Sie unter Auswertung der Nutzung von Amazon DocumentDB DocumentDB-Instances anhand von Metriken CloudWatch .

18. September 2024

Neues Feature

Amazon DocumentDB ist jetzt in der Region Afrika (Kapstadt) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

Amazon DocumentDB ist jetzt in der Region Europa (Spanien) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

17. September 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.10696)

Amazon DocumentDB unterstützt jetzt das Neuerstellen von Indizes mit `reIndex` in. `runCommand` Weitere Informationen finden Sie unter Indexverwaltung mit `reIndex`.

> **Note**
>
> `reIndex`wird nur auf Amazon DocumentDB 5.0 (Engine Patch Version 3.0.10696 und höher) unterstützt.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.10696) und Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.10898)

- `$setOnInsert`unterstützt das Feld jetzt bei Einfügungen, die aus Aktualisierungsvorgängen resultieren`_id`.
- Es wurde ein Problem behoben, das die Rückgewinnung von Speicherplatz nach einem Upgrade der Hauptversion auf Amazon DocumentDB 5.0 verhindert.

22. August 2024

Neues Feature

Amazon DocumentDB 5.0 (alle Engine-Patch-Versionen) und Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.5704)

Globale Amazon DocumentDB-Cluster unterstützen jetzt Cluster-Switchover und verwaltete Cluster-Failover. Weitere Informationen erhalten Sie unter Durchführen eines Switchovers für einen globalen Amazon DocumentDB-Cluster und Durchführen eines verwalteten Failovers für einen globalen Amazon DocumentDB-Cluster.

> **Note**
>
> Globale Cluster-Switchover und -Failover werden nur auf Amazon DocumentDB 4.0 und 5.0 unterstützt.

20. August 2024

Neues Feature

Für Amazon DocumentDB 3.6 (minimale Engine-Patch-Version 1.0.208662) erfordern TLS-CA-Zertifikatsaktualisierungen keinen Systemneustart mehr. Weitere Informationen finden Sie unter Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate.

8. August 2024

Neues Feature

Elastische Amazon DocumentDB-Cluster sind jetzt in den Regionen Asien-Pazifik (Hongkong), Kanada (Zentral) und Europa (Paris) verfügbar. Weitere Informationen finden Sie unter Regionale Verfügbarkeit für elastische Cluster.

23. Juli 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.8126) und Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.10709)

- Unterstützung für längere Indexnamen (bis zu 255 Zeichen) hinzugefügt. Weitere Informationen finden Sie unter Benennungseinschränkungen:.
- Das maximale Verbindungslimit wurde um das Zweifache erhöht:

Instance-Typ	Ursprüngliches Limit	Neues Limit
t3.medium	500	1000
t4g.medium	500	1000
r5.large	1700	3400
r5.xlarge	3500	7000
r5.2xlarge	7100	14200
r5.4xlarge	14200	28400
r5.8xlarge	28400	60000
r5.12xlarge	30000	60000
r5.16xlarge	30000	60000
r5.24xlarge	30000	60000
r6g.large	1700	3400
r6g.xlarge	3500	7000
r6g.2xlarge	7100	14200
r6g.4xlarge	14200	28400

Instance-Typ	Ursprüngliches Limit	Neues Limit
r6g.8xlarge	28400	60000
r6g.12xlarge	30000	60000
r6g.16xlarge	30000	60000

Weitere Informationen finden Sie unter Instance-Limits.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.8126)

Die Logik `ResumeToken` für die Synchronisation `CurrentTime` und den Change-Stream auf Lesegeräten wurde verbessert.

22. Juli 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.6742)

- Unterstützung für DML-Audit-Filterung hinzugefügt. Sie können jetzt Filterbedingungen einrichten, um DML-Auditprotokolle anhand ihrer spezifischen Anforderungen herauszufiltern, anstatt jede DML-Abfrage zu protokollieren. Weitere Informationen finden Sie unter DML-Auditereignisse filtern.
- Unterstützung bei der Dokumentenkomprimierung für Folgendes hinzugefügt:
  - Einstellung eines Mindestschwellenwerts für die Komprimierung
  - Aktivierung der Komprimierung für bestehende Sammlungen (gilt für neue Dokumente)
  - Lassen Sie die Standardkomprimierungseinstellung auf Clusterebene zu

  Weitere Informationen finden Sie unter Verwaltung der Dokumentenkomprimierung auf Sammlungsebene.
- Unterstützung für die Nutzung von Change-Streams auf Reader-Instances wurde hinzugefügt. Weitere Informationen finden Sie unter Verwendung von Change-Streams auf sekundären Instanzen.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.10593)

- Unterstützung für DML-Audit-Filterung hinzugefügt. Sie können jetzt Filterbedingungen einrichten, um DML-Auditprotokolle anhand ihrer spezifischen Anforderungen herauszufiltern, anstatt jede DML-Abfrage zu protokollieren. Weitere Informationen finden Sie unter DML-Auditereignisse filtern.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208662)

Die Indexbeschränkungen für db.r5.*- und db.r6.*-Instances in Amazon DocumentDB MVU wurden entfernt. Weitere Informationen finden Sie unter MVU-Voraussetzungen und Einschränkungen.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208662)

Amazon DocumentDB erkennt -NaN jetzt als gültiges JSON-Token.

9. Juli 2024

Neues Feature

Für Amazon DocumentDB 4.0 (minimale Engine-Patch-Version 2.0.10179) und 5.0 (minimale Engine-Patch-Version 3.0.4780) erfordern TLS-CA-Zertifikatsaktualisierungen keinen Systemneustart mehr. Weitere Informationen finden Sie unter Aktualisierung Ihrer Amazon DocumentDB-TLS-Zertifikate.

8. Juli 2024

Neues Feature

Elastische Amazon DocumentDB-Cluster sind jetzt in der Region Europa (Mailand) verfügbar. Weitere Informationen finden Sie unter Regionale Verfügbarkeit für elastische Cluster.

25. Juni 2024

Neues Feature

Die Authentifizierung mit AWS IAM ARNs ist in Amazon DocumentDB DocumentDB-Instance-basierten 5.0-Clustern in allen unterstützten Regionen verfügbar. Weitere Informationen finden Sie unter Authentifizierung mit IAM-Identität.

29. Mai 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.6742)

- Unterstützung für Operatoren und hinzugefügt. `$regexMatch` `$regexFind`
- Unterstützung hinzugefügt, um die volle Präzision der Audit-Logs bei der Adressierung großer Ganzzahlen zu gewährleisten. Audit-Logs behalten jetzt die exakte numerische Darstellung für alle Zahlen bei, wodurch jeglicher Genauigkeitsverlust vermieden wird.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.10593)

- Unterstützung hinzugefügt, um die volle Präzision der Audit-Logs bei der Adressierung großer Ganzzahlen sicherzustellen. Audit-Logs behalten jetzt die exakte numerische Darstellung für alle Zahlen bei, wodurch jeglicher Genauigkeitsverlust vermieden wird.

3. April 2024

Amazon DocumentDB ist jetzt in der Region Naher Osten (VAE) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.5721)

- Unterstützung für `bypassDocumentValidation` und detaillierte Fehlermeldungen für hinzugefügt. `$jsonSchema` Mehr über `bypassDocumentValidation` erfahren Sie unter bypassDocumentValidation.
- Unterstützung von `$expr` hinzugefügt.
- Unterstützung für unkorrelierte Verknüpfungen hinzugefügt. `$lookup`
- Unterstützung für die Beibehaltung von Validierungsregeln in der `$out` Aggregationsphase hinzugefügt.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.10392)

- Unterstützung für hinzugefügt. `bypassDocumentValidation` `$jsonSchema` Mehr über `bypassDocumentValidation` erfahren Sie unter bypassDocumentValidation.
- Unterstützung von hinzugefügt`$expr`.
- Unterstützung für unkorrelierte Verknüpfungen hinzugefügt. `$lookup`
- Unterstützung für die Beibehaltung von Validierungsregeln in der `$out` Aggregationsphase hinzugefügt.

Fehlerkorrekturen und andere Änderungen

- Ein Fehler beim Aufrufen `db.coll.stats()` auf der Mongo-Shell Version 1.7 und höher wurde behoben.
- Es wurde ein Problem mit Speicherlecks bei Change-Stream-Abfragen behoben, die `$regex` als Teil derselben Aggregationspipeline enthalten.

22. Februar 2024

Neue Features

Elastische Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster unterstützen jetzt die folgenden Funktionen:

- Lesbare sekundäre Shard-Instance-Replikate — weitere Informationen finden Sie in Schritt 5b von. Schritt 1: Erstellen Sie einen elastischen Cluster
- Cluster starten/beenden — weitere Informationen finden Sie unter. Stoppen und Starten eines elastischen Amazon DocumentDB-Clusters
- Konfigurierbare Shard-Instanzen — weitere Informationen finden Sie in Schritt 5b von. Schritt 1: Erstellen Sie einen elastischen Cluster
- Automatische Backups für Snapshots — weitere Informationen finden Sie unter. Verwaltung eines automatischen Backups eines Elastic Cluster-Snapshots
- Snapshot kopieren — weitere Informationen finden Sie unterEinen Elastic Cluster-Snapshot wird kopiert.

30. Januar 2024

Neue Features

Elastische Amazon DocumentDB-Cluster

Elastische Amazon DocumentDB-Cluster sind jetzt in den folgenden Regionen verfügbar:

- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Seoul)
- Südamerika (São Paulo)
- Europa (London)

Weitere Informationen finden Sie unter Elastische Cluster-Region und Versionsverfügbarkeit.

Globale Amazon DocumentDB-Cluster

Globale Cluster sind jetzt in beiden AWS GovCloud (US) Regionen verfügbar: AWS GovCloud (US-Ost) und AWS GovCloud (US-West).

10. Januar 2024

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Versionen 3.0.4574, 3.0.4780, 3.0.4960)

- Unterstützung für HNSW-Vektorindizes hinzugefügt. Weitere Informationen finden Sie unter Vektorsuche für Amazon DocumentDB.
- Unterstützung für Teilindizes hinzugefügt. Weitere Informationen finden Sie unter Teilweiser Index.
- Unterstützung für GC-Laufzeit für eine Sammlung innerhalb eines `current0p` Befehls hinzugefügt.
- Textindex-Unterstützung für die native Textsuche in Amazon DocumentDB hinzugefügt. Weitere Informationen finden Sie unter Textsuche mit Amazon DocumentDB durchführen.
- Unterstützung für `$jsonSchema` Schema-Schlüsselwörter`type`,,`allOf`,`oneOf`,`anyOf`,`not`,`maxItems`,`minItems`,`maxProperties`,`minPropertie` `patternProperties` `multipleOfdependencies`, und `uniqueItems` hinzugefügt.

Weitere Informationen finden Sie unter Verwenden der JSON-Schemavalidierung.

- Unterstützung für arithmetische Operatoren`$ceil`,,`$floor`,`$ln`, `$log` `$log10``$sqrt`, und hinzugefügt. `$exp`

  Weitere Informationen finden Sie unter Arithmetische Operatoren.
- Unterstützung für den Operator für bedingte Ausdrücke hinzugefügt. `$switch`
- Unterstützung für parallel `IVFFLAT` Vektorindex-Builds hinzugefügt. Die Dokumentation wurde aktualisiert, indem die Beschränkung für die Erstellung parallel `IVFFLAT` Vektorindizes aus dem Entwicklerhandbuch entfernt wurde.

Amazon DocumentDB 4.0 (Engine-Patch-Versionen 2.0.10124, 2.0.10179, 2.0.10221)

- Unterstützung für GC-Runtime in einer Sammlung innerhalb eines Befehls hinzugefügt. `currentOp`
- Unterstützung für `$jsonSchema` Schema-Schlüsselwörter`type`,`allOf`,`oneOf`,`anyOf`,`not`,`maxItems`,`minItems`,`maxProperties`,`minPropertie` `multipleOf``dependencies`, und hinzugefügt`uniqueItems`.

  Weitere Informationen finden Sie unter Verwenden der JSON-Schemavalidierung.
- Unterstützung für arithmetische Operatoren`$ceil`,,`$floor`,`$ln`, `$log` `$log10``$sqrt`, und hinzugefügt. `$exp`

  Weitere Informationen finden Sie unter Arithmetische Operatoren.
- Unterstützung für den Operator für bedingte Ausdrücke hinzugefügt. `$switch`

Fehlerkorrekturen und andere Änderungen

- Beim Aufrufen wurde eine Funktion hinzugefügt, bei der Groß- und Kleinschreibung nicht berücksichtigt wird. `db.runCommand("dbstats")` Kunden von Amazon DocumentDB 5.0 und 4.0 mit Engine-Patch-Versionen vor 3.0.4960 oder 2.0.10221 sollten diese neuesten Engine-Patches installieren.
- Ein Fehler beim Aufrufen `db.coll.stats()` auf der Mongo-Shell Version 1.7 und höher wurde behoben. Die Dokumentation wurde aktualisiert, indem der Tipp `db.coll.stats()` zur Fehlerbehebung bei Mongo-Shell aus dem Entwicklerhandbuch entfernt wurde.

20. Dezember 2023

Weitere Änderungen

Unterstützung für direkte Hauptversions-Upgrades in Amazon DocumentDB 3.6 und 4.0 aktiviert. Weitere Informationen finden Sie unter Direktes Upgrade der Hauptversion von Amazon DocumentDB.

13. Dezember 2023

Neue Features

Unterstützung für EC2 1-Klick-Konnektivität hinzugefügt. Weitere Informationen finden Sie unter Stellen Sie über Amazon eine Verbindung her EC2.

29. November 2023

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3727)

Neue Features

Unterstützung für die Vektorsuche hinzugefügt. Weitere Informationen finden Sie in diesem Blogbeitrag und Vektorsuche für Amazon DocumentDB im Amazon DocumentDB Developer Guide.

21. November 2023

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3727)

Neue Features

Unterstützung für I/O-optimierten Speicher hinzugefügt. Weitere Informationen finden Sie Amazon DocumentDB-Cluster-Speicherkonfigurationen im Amazon DocumentDB Developer Guide.

Integration für maschinelles Lernen ohne Code mit SageMaker Canvas hinzugefügt. Weitere Informationen finden Sie Maschinelles Lernen ohne Code mit Amazon SageMaker AI Canvas im Amazon DocumentDB Developer Guide.

17. November 2023

Neue Features

Amazon DocumentDB ist jetzt in der Region AWS GovCloud (USA-Ost) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208570)

Benutzerdefinierte lokale Variablennamen unterstützen jetzt „_“ (Unterstrich) für Projektionsoperatoren wie und. `$let` `$filter`

6. November 2023

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3727) und 4.0 (Engine-Patch-Version 2.0.9876)

Neue Features

- Unterstützung für `$jsonSchema` Schema-Schlüsselwörter`maxLength`,,,,, `minLength``maximum`, `minimum` und hinzugefügt. `exclusiveMaximum` `exclusiveMinimum` `items` `additionalItems`

  Bitte beachten Sie, dass die JSON-Schemavalidierung nur in instanzbasierten Clustern unterstützt wird.
- Unterstützung für den `$convert` Aggregationspipeline-Operator und seine abgeleiteten Kurzoperatoren`$toBool`,,,, `$toInt` `$toLong``$toDouble`, `$toString` und wurde hinzugefügt. `$toDecimal` `$toObjectId` `$toDate`
- Unterstützung für Mengenausdrucksoperatoren`$setDifference`, `$anyElementTrue` und wurde hinzugefügt. `$allElementTrue`

Fehlerkorrekturen und andere Änderungen

Es wurde ein Problem behoben, bei dem ein Change-Stream-Update von `-NaN` zu nicht angezeigt `NaN` wurde.

25. September 2023

Neue Features

Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Hongkong) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

20. September 2023

Neue Features

Unterstützung für direkte Hauptversions-Upgrades in Amazon DocumentDB 3.6 und 4.0 hinzugefügt. Weitere Informationen finden Sie unter Direktes Upgrade der Hauptversion von Amazon DocumentDB.

15. September 2023

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.3140) und 4.0 (Engine-Patch-Version 2.0.9686)

Unterstützung für den $jsonSchema-Schema-Validator wurde nur in instanzbasierten Clustern hinzugefügt. Weitere Informationen finden Sie unter Verwenden der JSON-Schemavalidierung.

11. September 2023

Neue Features

Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Hyderabad) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

3. August 2023

Neue Features

Elastische Amazon DocumentDB-Cluster

- Amazon DocumentDB Elastic-Cluster unterstützen jetzt die folgenden Operationen:
  - `top`
  - `collStats`
  - `hint`
  - `dataSize`

  Die Unterstützte MongoDB APIs, Operationen und Datentypen in Amazon DocumentDB vollständige Liste der unterstützten Befehle und Operationen finden Sie unter.
- Time to Live (TTL) -Indizes werden jetzt unterstützt.
- Indizes `hints` werden jetzt mit Indexausdrücken unterstützt.

13. Juli 2023

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.1948)

- Unterstützung für die Komprimierung von Dokumenten hinzugefügt.
- Unterstützung für parallel Indexerstellungen hinzugefügt.
- Unterstützung für den Status der Indexerstellung hinzugefügt.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.9259)

- Unterstützung für parallel Indexerstellungen hinzugefügt.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.1948)

- Authentifizierungsproblem `createCollection` bei elastischen Amazon DocumentDB-Clustern behoben, wenn Benutzer keinen Zugriff auf Systemsammlungen hatten.
- Es wurde ein Problem behoben, bei dem Instances in der sekundären Region nicht dieselben Instance-Namen für die primäre Region verwenden konnten.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.9259)

- Das Hinzufügen interner Überwachungsanfragen zu den Audit-Protokollen wurde beendet.

7. Juni 2023

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 5.0

- r5- und t3.medium-Instances werden jetzt in Amazon DocumentDB 5.0 unterstützt.
- `engineVersion`Die Standardoption befindet sich `5.0.0` im AWS SDK,, und. AWS CLI AWS CloudFormation

10. Mai 2023

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.1361)

- Unterstützung für `ignoreunknownindexoptions` im Befehl hinzugefügt. createIndex
- Das Hinzufügen interner Überwachungsanfragen zu den Überwachungsprotokollen wurde beendet.
- Benutzerdefinierte lokale Variablennamen unterstützen jetzt „_“ (Unterstrich) für Projektionsoperatoren wie `$let` und`$filter`.

4. April 2023

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.8934)

- Das Problem mit der DML-Überwachung, wenn sie während einer laufenden Arbeitslast aktiviert wurde, wurde behoben.
- Es wurde ein Problem mit der DML-Überwachung behoben, bei dem aggregierten Befehlen mit einem Hinweis ein Zeichenfolgenwert übergeben wurde.
- Es wurde ein Problem behoben, bei dem der `listCollections` Befehl nicht funktionierte, wenn Benutzer mit der Rolle readwriteanydatabase sowohl die Optionen AuthorizedCollections als auch NameOnly auf true gesetzt hatten.

- Das Problem, dass numerische Zeichenketten in einem Feldnamen korrekt analysiert wurden, wurde behoben.
- Brechen Sie lange laufende Cursor ab, wenn sie sich auf die Speicherbereinigung auswirken.
- Benutzerdefinierte lokale Variablennamen unterstützen jetzt „_“ (Unterstrich) für Projektionsoperatoren wie und. `$let` `$filter`

22. März 2023

Neue Features

Elastische Amazon DocumentDB-Cluster sind jetzt in den Regionen Asien-Pazifik (Singapur), Asien-Pazifik (Sydney) und Asien-Pazifik (Tokio) verfügbar. Weitere Informationen finden Sie unter Elastische Cluster-Region und Versionsverfügbarkeit.

1. März 2023

Neue Features

Amazon DocumentDB 5.0 (Engine-Patch-Version 3.0.775)

- Amazon DocumentDB 5.0 eingeführt
  - MongoDB 5.0-Kompatibilität (Unterstützung für MongoDB 5.0-API-Treiber)
  - Support für clientseitige Verschlüsselung auf Feldebene (FLE). Sie können jetzt Felder auf der Clientseite verschlüsseln, bevor Sie die Daten in den Amazon DocumentDB-Cluster schreiben. Weitere Informationen finden Sie unter Clientseitige Verschlüsselung auf Feldebene
  - Neue Aggregationsoperatoren:, `$dateAdd` `$dateSubtract`
- Das Speicherlimit wurde für alle instanzbasierten Amazon DocumentDB-Cluster und shard-basierten Elastic-Cluster auf 128 TiB erhöht.
- Amazon DocumentDB 5.0 unterstützt jetzt Indexscans mit dem `$elemMatch` Operator in der ersten Verschachtelungsebene. Indexscans werden unterstützt, wenn Abfragen nur eine `$elemMatch` Filterebene haben und die verschachtelte `$elemMatch` Abfrage keinen Indexscan unterstützt.

  Abfrageform, die den Indexscan unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": "xyz", "c": "abc"} } })
```

Abfrageform, die den Indexscan nicht unterstützt:

```
db.foo.find( { "a": {$elemMatch: { "b": {$elemMatch: { "d": "xyz", "e":
 "abc"} }} } })
```

27. Februar 2023

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 4.0

Unterstützung für AWS Lambda hinzugefügt. Weitere Informationen finden Sie unter Verwendung AWS Lambda mit Change-Streams.

2. Februar 2023

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208432)

- Es wurde ein Problem mit der DML-Überwachung behoben, wenn sie während einer laufenden Arbeitslast aktiviert wurde.
- Es wurde ein Problem mit der DML-Überwachung behoben, bei dem aggregierten Befehlen mit einem Hinweis ein Zeichenfolgenwert übergeben wurde.
- Es wurde ein Problem behoben, bei dem der `listCollections` Befehl nicht funktionierte, wenn Benutzer mit der Rolle readwriteanydatabase sowohl die Optionen AuthorizedCollections als auch NameOnly auf true gesetzt hatten.
- Das Problem, dass numerische Zeichenketten in einem Feldnamen korrekt analysiert wurden, wurde behoben.
- Brechen Sie lange laufende Cursor ab, wenn sie sich auf die Speicherbereinigung auswirken.

30. November 2022

Neue Features

Elastische Amazon DocumentDB-Cluster

Amazon DocumentDB Elastic Clusters ist ein neuer Typ von Amazon DocumentDB-Cluster, mit dem Benutzer das MongoDB-Sharding nutzen können, um ihren Cluster APIs zu skalieren. Elastic Cluster verarbeiten praktisch jede Anzahl von Lese- und Schreibvorgängen mit Petabyte an Speicherkapazität, indem sie die Daten und Rechenleistung auf mehrere zugrunde liegende Recheninstanzen und Volumes verteilen. Weitere Informationen finden Sie unter Verwenden von elastischen Amazon DocumentDB-Clustern.

09. August 2022

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208152) und 4.0

- Unterstützung für den Datentyp Decimal128 wurde hinzugefügt. Decimal128 ist ein BSON-Datentyp, der in allen Regionen unterstützt wird, in denen DocumentDB verfügbar ist.

  Weitere Informationen finden Sie unter Data Types.
- Unterstützung für die Prüfung von DML-Abfragen mit Amazon CloudWatch Logs hinzugefügt. Amazon DocumentDB kann jetzt DML-Ereignisse (Data Manipulation Language) und DDL-Ereignisse (Data Definition Language) in Amazon Logs aufzeichnen. CloudWatch

  Weitere Informationen finden Sie in diesem Blogbeitrag.

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.208152) und 4.0

- Sie können jetzt Ihr eigenes Passwort mit einem eigenen Passwort mit Privilegien ändern. `changeOwnPassword`

25. Juli 2022

Neue Features

Amazon DocumentDB 4.0

Sie können jetzt schneller Cluster erstellen, da Sie Klone erstellen können, die dasselbe DocumentDB-Cluster-Volume verwenden und dieselben Daten wie der ursprüngliche Cluster haben. Einzelheiten finden Sie unter Amazon DocumentDB-Cluster verwalten.

27. Juni 2022

Neue Features

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.7509)

Amazon DocumentDB passt die Größe Ihrer Datenbank dynamisch auf der Grundlage von Nutzungsmustern an. Durch das Hinzufügen weiterer Daten wird der Speicherplatz auf bis zu 64 Tebibyte (TiB) erhöht, und durch das Löschen von Daten wird der zugewiesene Speicherplatz verringert.

29. April 2022

Neue Features

Amazon DocumentDB ist jetzt in der Region China (Peking) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

7. April 2022

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Versionen 1.0.207836 und 1.0.208015) und 4.0 (Engine-Patch-Versionen 2.0.6142 und 2.0.6948)

Amazon DocumentDB Performance Insights ist jetzt als Vorschauversion verfügbar. Sie können jetzt den Leistungsverlauf von sieben Tagen ohne zusätzliche Kosten in einem sich öffnenden Fenster speichern. Weitere Informationen finden Sie unter Monitoring with Performance Insights.

16. März 2022

Neue Features

Amazon DocumentDB ist jetzt in der Region Europa (Mailand) verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

8. Februar 2022

Neue Features

Amazon DocumentDB R6g- und T4g-Instances sind jetzt im asiatisch-pazifischen Raum, Südamerika und Europa verfügbar. Weitere Informationen finden Sie in diesem Blogbeitrag.

24. Januar 2022

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207684) und 4.0 (Engine-Patch-Version 2.0.5170)

- Amazon DocumentDB bietet jetzt eine kostenlose Testversion. Einzelheiten finden Sie auf der Seite mit der kostenlosen Testversion von Amazon DocumentDB.
- Sie können jetzt erweiterte Funktionen mit Geospatial-Abfragen verwenden, darunter die folgenden: APIs
  - `$geoWithin`
  - `$geoIntersects`
- Unterstützung für die folgenden MongoDB-Operatoren hinzugefügt:
  - `$mergeObjects`
  - `$reduce`

  Weitere Informationen finden Sie unter Abfragen von Geodaten mit Amazon DocumentDB.

21. Januar 2022

Neue Features

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.5706)

- Amazon DocumentDB Graviton2-Instances (r6g.large, r6g.2xlarge, r6g.4xlarge, r6g.8xlarge, r6g.12xlarge, r6g.16xlarge und t4g.medium) werden jetzt unterstützt.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207781) und 4.0 (Engine-Patch-Version 2.0.5706)

- Unterstützung für die folgende MongoDB APIs wurde hinzugefügt:
  - `$reduce`
  - `$mergeObjects`
  - `$geoWithin`
  - `$geoIntersects`

25. Oktober 2021

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207780) und 4.0 (Engine-Patch-Version 2.0.5704)

- Unterstützung für die folgende MongoDB hinzugefügt APIs
  - `$literal`
  - `$map`
  - `$$ROOT`
- Support für GeoSpatial Abfragefunktionen. Weitere Informationen finden Sie in diesem Blogbeitrag
- Support für die Zugriffskontrolle mit benutzerdefinierten Rollen. Weitere Informationen finden Sie in diesem Blogbeitrag
- Amazon DocumentDB JDBC-Treiber zur Aktivierung der Konnektivität von BI-Tools wie Tableau und Abfragetools wie SQL Workbench

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207780) und 4.0 (Engine-Patch-Version 2.0.5704)

- Fehlerkorrektur für die korrekte Sortierung`$natural`, wenn ein explizites Zeichen zusammen mit `.sort()` `$natural`
- Bugfix für den Change-Stream, mit dem gearbeitet werden soll `$redact`
- Bugfix für `$ifNull` das Arbeiten mit einem leeren Array
- Bugfix für übermäßigen consumption/server Ressourcenabsturz, wenn ein aktuell angemeldeter Benutzer gelöscht wird oder diesem Benutzer die Rechte für eine laufende Aktivität entzogen werden
- Fehlerbehebung bei der Anmeldung und Überprüfung der Rechte `listDatabase` `listCollection`
- Fehler behoben: Deduplizierungslogik für Elemente mit mehreren Schlüsseln

24. Juni 2021

Neue Features

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207117) und 4.0 (Engine-Patch-Version 2.0.3371)

- Die Instances r5.8xlarge und r5.16xlarge werden jetzt unterstützt. Weitere Informationen finden Sie im Blogbeitrag Amazon DocumentDB unterstützt jetzt r5.8xlarge- und r5.16xlarge-Instances.
- Globale Cluster werden jetzt unterstützt, um eine Notfallwiederherstellung nach regionalen Ausfällen zu ermöglichen und globale Lesevorgänge mit geringer Latenz zu ermöglichen, indem Lesevorgänge vom nächstgelegenen Amazon DocumentDB-Cluster zugelassen werden. Beachten Sie, dass globale Cluster derzeit in den Regionen Südamerika (São Paulo), Europa (Mailand), China (Peking) und China (Ningxia) nicht unterstützt werden.

4. Mai 2021

Neue Features

In diesem Blogbeitrag finden Sie alle neuen Funktionen.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207117) und 4.0 (Engine-Patch-Version 2.0.3371)

- `renameCollection`
- `$zip`
- `$indexOfArray`
- `$reverseArray`
- `$natural`
- `$hint`Unterstützung für Updates
- Index-Scan für `distinct`

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.207117) und 4.0 (Engine-Patch-Version 2.0.3371)

- `$in`Reduzierter Speicherverbrauch für Abfragen
- Ein Speicherleck in Multikey-Indizes wurde behoben
- Der Explain-Plan und die Profiler-Ausgabe für wurden behoben `$out`
- Es wurde ein Timeout für Operationen aus dem internen Überwachungssystem hinzugefügt, um die Zuverlässigkeit zu verbessern
- Es wurde ein Fehler behoben, der sich auf die an Multikey-Indizes übergebenen Abfrageprädikate auswirkte

15. Januar 2021

Neue Features

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.722)

- Keine

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Fähigkeit, einen Index in der Aggregationsphase zu verwenden `$lookup`

- `find()`Abfragen mit Prognosen können direkt von einem Index aus bedient werden (abgedeckte Abfrage)
- Fähigkeit zur Verwendung `hint()` mit dem `findAndModify`
- Leistungsoptimierungen für `$addToSet` den Bediener
- Verbesserungen zur Reduzierung der allgemeinen Indexgrößen
- Neue Aggregationsoperatoren:`$ifNull`,`$replaceRoot`,`$setIsSubset`, `$setInstersection``$setUnion`, und `$setEquals`
- Benutzer können auch ihre eigenen Cursor beenden, ohne die Rolle zu benötigen `KillCursor`

9. November 2020

Neue Features

Alle neuen Funktionen finden Sie in diesem Blogbeitrag.

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.722)

- MongoDB 4.0-Kompatibilität
- ACID-Transaktionen
- Support für `cluster(client.watch()` oder `mongo.watch())` und die `(db.watch())` Change-Streams auf Datenbankebene
- Möglichkeit, einen Change-Stream zu starten oder fortzusetzen mit `startAtOperationTime`
- Verlängern Sie die Aufbewahrungsfrist für Ihren Change-Stream auf 7 Tage (zuvor 24 Stunden)
- AWS DMS Ziel für Amazon DocumentDB 4.0
- CloudWatch Metriken: `TransactionsOpenTransactionsOpenMax`, `TransactionsAbortedTransactionsStarted`, und `TransactionsCommitted`
- Neue Felder für Transaktionen in `currentOpServerStatus`, und`profiler`.
- Möglichkeit, einen Index in der `$lookup` Aggregationsphase zu verwenden
- `find()`Abfragen mit Prognosen können direkt von einem Index aus bedient werden (abgedeckte Abfrage)
- Fähigkeit zur Verwendung `hint()` mit dem `findAndModify`
- Leistungsoptimierungen für `$addToSet` den Bediener
- Verbesserungen zur Reduzierung der allgemeinen Indexgrößen.

- Neue Aggregationsoperatoren:`$ifNull`,`$replaceRoot`,`$setIsSubset`, `$setInstersection``$setUnion`, und `$setEquals`
- Mit den `ListDatabase` Befehlen `ListCollection` und können Sie nun optional die `authorizedDatabases` Parameter `authorizedCollections` und verwenden, um Benutzern zu ermöglichen, die Sammlungen und Datenbanken aufzulisten, auf die sie zugreifen dürfen, ohne die `listCollections` jeweiligen `listDatabase` Rollen und
- Benutzer können auch ihre eigenen Cursor beenden, ohne die `KillCursor` Rolle zu benötigen
- Der Vergleich numerischer Typen von Filialdokumenten entspricht jetzt dem Vergleich numerischer Typen von Dokumenten der ersten Ebene. Das Verhalten in Amazon DocumentDB 4.0 ist jetzt mit MongoDB kompatibel.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Keine

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 4.0 (Engine-Patch-Version 2.0.722)

- `$setOnInserter`laubt keine Aktualisierungen mehr, wenn der Positionsoperator verwendet wird. $ Das Verhalten in Amazon DocumentDB 4.0 ist jetzt mit MongoDB kompatibel.
- Problem mit `$createCollection` und Set behoben `autoIndexId`
- Projektion für verschachtelte Dokumente
- Die Standardeinstellung für den Arbeitsspeicher wurde so geändert, dass sie mit der Größe des Instanzspeichers skaliert
- Verbesserungen bei der Müllabfuhr
- Suche mit leerem Pfadschlüssel, Verhaltensunterschied bei Mongo
- `dateToString`Fehler im Verhalten der Zeitzone behoben
- Behoben `$push` (Aggregation), um die Sortierreihenfolge zu respektieren
- Ein Fehler im Aggregat `$currentOp` wurde behoben
- Problem mit einem `readPreference` Sekundärgerät behoben
- Es wurde ein Problem mit der Überprüfung behoben, `$createIndex` dass es sich um dieselbe Datenbank handelt, in der der Befehl ausgegeben wurde

- Inkonsistentes Verhalten für`minKey`, Suche schlägt fehl `maxKey`
- Es wurde ein Problem behoben, bei dem der `$size` Operator nicht mit einem zusammengesetzten Array funktionierte
- Problem mit der Negation von `$in` mit Regex behoben
- Es wurde ein Problem behoben, bei dem der `$distinct` Befehl für eine Ansicht ausgeführt wurde
- Es wurde ein Problem mit Aggregationen und Suchbefehlen behoben, bei denen fehlende Felder unterschiedlich sortiert wurden
- Problem behoben`$eq`, bei dem reguläre Ausdrücke den Typ nicht überprüfen
- Ein `$currentDate` Fehler im Verhalten bei der Ordinalposition von Zeitstempeln wurde behoben
- Die Granularität im Millisekundenbereich wurde behoben für `$currentDate`

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Keine

30. Oktober 2020

Neue Features

Alle neuen Funktionen finden Sie in diesem Blogbeitrag.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Es wurde die Möglichkeit hinzugefügt, einen Change-Stream-Cursor auf Cluster-Ebene `(client.watch()` oder in der Datenbank zu öffnen `mongo.watch()) (db.watch())`
- Möglichkeit, die Aufbewahrungsfrist für Change-Streams auf 7 Tage (zuvor 24 Stunden) zu erhöhen

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Verschiedene allgemeine Verbesserungen der Fallleistung
- Eine gezielte Verbesserung der Sicherheit

- Es wurde ein Problem behoben, bei dem die Sortierung im zweiten Feld eines zusammengesetzten Indexes übersprungen wurde
- Aktiviert den regulären Index für Gleichheit für ein einzelnes Feld eines Indexes mit mehreren Schlüsseln (nicht zusammengesetzt)
- Die Bedingung für das Autorisierungsrennen wurde behoben
- Es wurde ein Problem behoben, das zu einem seltenen Absturz bei der Garbage-Collection führte
- Verbesserung der RBAC-Sicherheit
- Metrik hinzugefügt `databaseConnectionsMax`
- Leistungsverbesserungen für bestimmte Workloads auf Instanzen `r5.24xlarge`

22. September 2020

Neue Features

Alle neuen Funktionen finden Sie in diesem Blogbeitrag.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- `$out`Aggregationsphase
- Die maximale Anzahl von Verbindungen und Cursor pro Instanz wurde um das Zehnfache erhöht

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Keine

10. Juli 2020

Neue Features

Alle neuen Funktionen finden Sie in diesem Blogbeitrag.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Regionsübergreifende Snapshot-Kopie

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Keine

30. Juni 2020

Neue Features

Alle neuen Funktionen finden Sie in diesem Blogbeitrag.

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Mittlere T3-Instanzen

Fehlerkorrekturen und andere Änderungen

Amazon DocumentDB 3.6 (Engine-Patch-Version 1.0.206295)

- Rückgewinnung von Arbeitsspeicher im Leerlauf für T3-Instances
- Verbesserungen bei der Authentifizierung
- Verbesserte Leistung der SASL-Authentifizierung
- `currentOp`Problem beim Überschreiten der maximal möglichen Operationen behoben
- `killOps`Problem beim Bulk-Update und Löschen behoben
- `$sample`Leistungsverbesserungen mit `$match`
- Die Unterstützung für „$$In Cond Case“ in der Redact-Phase wurde behoben
- Verschiedene wiederkehrende Ursachen für Abstürze wurden behoben
- Verbesserungen beim TTL-Sweeping zur Reduzierung der Latenz IOs
- Optimierte Speichernutzung für `$unwind`
- Es wurde ein Fehler behoben, bei dem das Rennen mit Drop-Index angezeigt wurde
- Die Rennbedingung bei der gleichzeitigen Indexerstellung wurde behoben
- Seltener Absturz im `hash_search` Index behoben

Dokumentenverlauf für den Amazon DocumentDB Developer Guide

- API-Version: 2014-10-31
- Letzte Aktualisierung der Dokumentation: 23. Juli 2024

In der folgenden Tabelle wird die Dokumentation für diese Version des Amazon DocumentDB Developer Guide beschrieben.

Änderung	Beschreibung	Datum
AWS verwaltete Richtlinienaktualisierung — Änderung der Richtlinie	Amazon DocumentDB aktualisiert die Vollzugriffsrichtlinien für elastische Cluster.	11. Februar 2025
AWS verwaltetes Richtlinienupdate — Änderung der Richtlinie	Amazon DocumentDB aktualisiert die Vollzugriffsrichtlinien für elastische Cluster.	21. Februar 2024
AWS verwaltetes Richtlinienupdate — Änderung der Richtlinie	Amazon DocumentDB aktualisiert die Richtlinien für Lesezugriff und vollen Zugriff für elastische Cluster.	21. Juni 2023
AWS verwaltetes Richtlinienupdate — neue Richtlinie	Amazon DocumentDB führt eine neue Nur-Lese-Richtlinie für elastische Cluster ein.	08. Juni 2023
AWS verwaltetes Richtlinienupdate — neue Richtlinie	Amazon DocumentDB führt eine neue Vollzugriffsrichtlinie für elastische Cluster ein.	5. Juni 2023
MongoDB 5.0-Kompatibilität	Amazon DocumentDB ist jetzt mit Version 5.0 von MongoDB kompatibel.	1. März 2023

Aktualisierung der Richtlinien	Um die Elastic-Cluster-Funktion von Amazon DocumentDB zu unterstützen, wird die AmazonDocDBConsole FullAccess Richtlinie aktualisiert und die AmazonDoc ElasticServiceRolePolicy DB-Funktion eingeführt.	30. November 2022
Elastische Cluster	Neue Elastic Cluster-Funktion hinzugefügt, die Hash-basierte Partitionierung (Sharding) von Daten im verteilten Speichersystem von Amazon DocumentDB unterstützt.	30. November 2022
Globale Cluster	Dokumentation zur Verwendung von Global Clusters hinzugefügt.	2. Juni 2021
Abonnements für Veranstaltungen	Dokumentation für Event-Abonnements hinzugefügt.	26. März 2021
Aktualisierungen auf Version 3.6	Dokumentierte Verbesserungen gegenüber Version 3.6 in Bezug auf rollenbasierte Zugriffskontrollen, Aggregationsoperatoren und Leistung.	15. Januar 2021
MongoDB 4.0-Kompatibilität	Amazon DocumentDB ist jetzt mit Version 4.0 von MongoDB kompatibel.	9. November 2020

Anleitungen für die ersten Schritte	Neue Anleitungen für die ersten Schritte mit Amazon DocumentDB mithilfe von Amazon AWS Cloud9 EC2, Robo3T oder Studio3T.	15. August 2020
Zusätzliche Availability Zones werden unterstützt	Amazon DocumentDB hat Unterstützung für eine zusätzliche Availability Zone im asiatisch-pazifischen Raum (Seoul) hinzugefügt (ap-north east-2).	14. Juli 2020
Unterstützung für das Kopieren von Schnappsc hüssen zwischen Regionen hinzugefügt.	Amazon DocumentDB hat Unterstützung für das Kopieren von Cluster-S napshots hinzugefügt. AWS-Regionen Weitere Informationen finden Sie unter Kopieren von Snapshots zwischen Regionen.	10. Juli 2020
Unterstützung für die Instanzkl asse T3 wurde hinzugefügt.	Unterstützung für T3-Instan ce-Typen in allen Regionen hinzugefügt, die Amazon DocumentDB unterstüt zen. Weitere Informationen finden Sie unter Unterstützte Instance-Klassen nach Region und Instance-Klassensp ezifikationen.	30. Juni 2020
Unterstützung für hinzugefügt AWS GovCloud (US).	Amazon DocumentDB ist jetzt in der AWS GovCloud (US) Region (us-gov-west-1) verfügbar.	29. Juni 2020

16 neue CloudWatch Metriken wurden hinzugefügt.	Amazon DocumentDB hat Unterstützung für 16 neue CloudWatch Amazon-Metriken hinzugefügt. Weitere Informationen finden Sie unter Amazon DocumentDB überwachen mit CloudWatch.	23. Juni 2020
Unterstützung für Null-Zeichen und den $regex-Operator wurde hinzugefügt.	Amazon DocumentDB hat Unterstützung für Nullzeichen in Zeichenketten und die Möglichkeit hinzugefügt, einen Index für $regex zu verwenden. Informationen zu den unterstützten MongoDB APIs - und Aggregationspipeline-Funktionen für Amazon DocumentDB finden Sie unter Funktionale Unterschiede mit MongoDB.	22. Juni 2020
Unterstützung für verbesserte Multi-Key-Indexierungsfunktionen hinzugefügt.	Amazon DocumentDB hat Unterstützung für verbesserte Multi-Key-Indexierungsfunktionen hinzugefügt, darunter die Indizierung von Arrays mit mehr als 2.048 Byte und die Möglichkeit, einen zusammengesetzten Multi-Key-Index mit mehreren Schlüsseln im selben Array zu erstellen. Weitere Informationen finden Sie unter Funktionsunterschiede bei MongoDB.	23. April 2020

Unterstützung für den Löschschutz für einen Amazon DocumentDB AWS CloudFormation DocumentDB-Stack hinzugefügt.	Amazon DocumentDB hat Unterstützung für die Aktivierung des Löschschutzes bei der Erstellung eines Amazon DocumentDB AWS CloudFormation DocumentDB-Stacks hinzugefügt.	20. April 2020
Unterstützung für rollenbasierte Zugriffskontrolle hinzugefügt.	Amazon DocumentDB hat Unterstützung für die rollenbasierte Zugriffskontrolle mithilfe integrierter Rollen hinzugefügt.	26. März 2020
Unterstützung für eine zusätzliche Availability Zone in Kanada (Central) (ca-central-1) hinzugefügt.	Amazon DocumentDB ist jetzt in der Region Kanada (Central) (ca-central-1) mit Instances der Klasse R5 und 3 Availability Zones verfügbar.	26. März 2020
Unterstützung für zwei weitere MongoDB APIs hinzugefügt.	Amazon DocumentDB hat Unterstützung für `$dateFromString` und `executionStats` APIs MongoDB hinzugefügt.	23. März 2020
Unterstützung für fünf weitere MongoDB APIs hinzugefügt.	Amazon DocumentDB hat Unterstützung für`$objectToArray` ,, `$arrayToObject` `$slice` `$mod`, und `$range` APIs MongoDB hinzugefügt.	6. Februar 2020
Unterstützung für Kanada (Central) hinzugefügt.	Amazon DocumentDB ist jetzt in der Region Kanada (Central) (ca-central-1) mit Instances der Klasse R5 verfügbar.	11. Dezember 2019

ChangeStreamLogSizeUnterstützung für hinzugefügt.	Amazon DocumentDB hat Unterstützung `ChangeStreamLogSize` für Cloudwatch-Metriken hinzugefügt.	22. November 2019
Unterstützung für die Region Europa (Paris) hinzugefügt	Amazon DocumentDB ist jetzt in der Region Europa (Paris) (eu-west-3) mit Instances der Klasse R5 verfügbar.	30. Oktober 2019
Unterstützung für die Region Asien-Pazifik (Mumbai) wurde hinzugefügt	Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Mumbai) (ap-south-1) mit Instances der Klasse R5 verfügbar.	17. Oktober 2019
Unterstützung für drei weitere MongoDB hinzugefügt APIs	Amazon DocumentDB hat Unterstützung für `$addFields` `$concatArrays` , und `$lookup` APIs MongoDB hinzugefügt.	16. Oktober 2019
Unterstützung für die Region Asien-Pazifik (Singapur) hinzugefügt	Amazon DocumentDB ist jetzt in der Region Asien-Pazifik (Singapur) (ap-southeast-1) mit Instances der Klasse R5 verfügbar.	14. Oktober 2019
Neues Dokument für die Aktualisierung von TLS-Zertifikaten hinzugefügt	Anweisungen zum Aktualisieren von Zertifizierungsstellenzertifikaten zur Verwendung des neuen Zertifizierungsstellenzertifikats zum Erstellen von TLS-Verbindungen wurden hinzugefügt.	2. Oktober 2019

API-Unterstützung für Zertifikate hinzugefügt	Amazon DocumentDB ist ein neuer Zertifikatsdatentyp für Instances. Weitere Informationen finden Sie unter DBInstance.	1. Oktober 2019
Support für die Erstellung von Abfrageprofilen	Amazon DocumentDB hat die Möglichkeit hinzugefügt, unterstützte Operationen auf den Instances und Datenbanken Ihres Clusters zu profilieren.	19. August 2019
Dritte AZ im asiatisch-pazifischen Raum (Tokio) hinzugefügt	Amazon DocumentDB hat eine dritte Availability Zone (AZ) für Ihre Compute-Instances im asiatisch-pazifischen Raum (Tokio) hinzugefügt.	9. August 2019

Support für zusätzliches Mongo APIs	Unterstützung für zusätzlic he Funktionen der Aggregati onspipeline hinzugefügt, zu denen die Aggregati onsoperatoren `$in` `$isoWeek` `$isoWeekY ear` ,`$isoDayOfWeek` ,, und die `$dateToString` `$addToSet` Aggregati onsphase gehören. Amazon DocumentDB hat außerdem Unterstützung für den `top()` Befehl zur Diagnose auf Sammlungsebene und die Möglichkeit hinzugefügt, den `expireAfterSeconds` Parameter für TTL-Indizes mithilfe des Befehls zu ändern. `collMod()`	31. Juli 2019
Unterstützung für Europa (London) hinzugefügt	Amazon DocumentDB ist jetzt in Europa (London) (eu-west-2) mit Instances der Klasse R5 verfügbar.	18. Juli 2019
Codebeispiele wurden hinzugefügt	In R und Ruby wurden Codebeispiele für die programmgesteuerte Verbindung zu Amazon DocumentDB hinzugefügt.	17. Juli 2019
Bewährtes Verfahren hinzugefügt	Es wurde eine Best Practice hinzugefügt, die Ihnen hilft, Ihre Amazon DocumentD B DocumentDB-Kosten zu verwalten.	17. Juli 2019

Support für das Stoppen und Starten eines Clusters	Amazon DocumentDB hat Unterstützung für das Stoppen und Starten von Clustern hinzugefügt, um die Kosten für Entwicklungs- und Testumgebungen zu verwalten.	1. Juli 2019
Support für den Schutz vor Clusterlöschung	Um Ihre Cluster vor versehentlichem Löschen zu schützen, hat Amazon DocumentDB einen Löschschutz hinzugefügt. Weitere Informationen finden Sie in den folgenden Themen: Erstellen eines Amazon DocumentDB-Clusters, Ändern eines Amazon DocumentDB-Clusters, Löschen eines Amazon DocumentDB-Clusters und `DeletionProtection` im API-Thema. DBCluster	1. Juli 2019
Aktualisierung der funktionalen Unterschiede	Implizite Transaktionen wurden den Funktionsunterschieden hinzugefügt.	26. Juni 2019
Hinzufügung funktionaler Unterschiede	Hinweis zur Speicherung und Indexkomprimierung in Amazon DocumentDB hinzugefügt.	13. Juni 2019
Zusätzliche Region wird unterstützt	Amazon DocumentDB ist jetzt im asiatisch-pazifischen Raum (Sydney) (ap-southeast-2) mit Instances der Klasse R5 verfügbar.	5. Juni 2019

Die R5-Instance-Klasse wird in weiteren Regionen unterstützt	Unterstützung für R5-Instance-Klasse für 4 zusätzliche Regionen hinzugefügt: USA Ost (Ohio), USA Ost (Nord-Virginia), USA West (Oregon) und EU (Irland). Mit dieser Änderung werden R5-Instances in allen Regionen unterstützt, die Amazon DocumentDB unterstützen.	17. Mai 2019
Zusätzliche Regionen werden unterstützt	Unterstützung für zwei weitere Regionen, Asien-Pazifik (Tokio) (ap-northeast-1) und Asien-Pazifik (Seoul) (ap-northeast-2), mit R5-Instance-Klassen hinzugefügt. Weitere Informationen finden Sie unter Unterstützte Instance-Klassen nach Region und Instance-Klassen-Spezifikationen.	8. Mai 2019
Weitere Beispiele für Verbindungscodes wurden hinzugefügt	Es wurden Codebeispiele in Java und C# für die Verbindung mit Amazon DocumentDB hinzugefügt.	24. April 2019

Zusätzliche Mongo-API-Unterstützung	Es wurde die Unterstützung von sieben Aggregations-Zeichenfolgenoperatoren (`$indexOfBytes`, `$indexOfCP`, `$strLenBytes`, `$strLenCP`, `$toLower`, `$toUpper`, `$split`), neun Datumszeitoperatoren (`$dayOfYear`, `$dayOfMonth`, `$dayOfWeek`, `$year`, `$month`, `$hour`, `$minute`, `$second`, `$millisecond`) und die Aggregations-Pipeline-Stufe `$sample` hinzugefügt.	4. April 2019
Beispiele für Verbindungscodes hinzugefügt	Es wurden Codebeispiele in Python, Node.js, PHP und Go für die Verbindung mit Amazon DocumentDB hinzugefügt.	21. März 2019
Support für die Region Frankfurt und R5-Instances	Unterstützung für die Region Europa (Frankfurt) (eu-central-1) mit R5-Instance-Klassen hinzugefügt. Weitere Informationen finden Sie unter Unterstützte Instance-Klassen nach Region und Instance-Klassen-Spezifikationen.	13. März 2019

Unterstützung von Aggregationspipeline-Operatoren	Es wurde die Unterstützung für neue Aggregations-Zeichenfolgenoperatoren (`$concat`, `$substr`, `$substrBytes` , `$substrCP` , `$strcasecmp`), einen Array-Aggregationsoperator (`$size`), einen Aggregationsgruppen-Akkumulatoroperator (`$push`) und Aggregationsstufen (`$redact` und `$indexStats`) hinzugefügt. Wir haben außerdem die Unterstützung für Positional-Array-Operatoren (`$[]` und `$[<identifier>]`) und `hint()` hinzugefügt.	28. Februar 2019
Motor-Upgrades	Es wurde eine Dokumentation zur Ermittlung ausstehender Cluster-Änderungen und Aktualisierung der Engine Version Ihres Clusters hinzugefügt.	15. Februar 2019
Ereignisse prüfen	Unterstützung für die Prüfung von Datenbankereignissen mit Amazon CloudWatch Logs hinzugefügt.	12. Februar 2019
Schnellstart	Es wurde ein Schnellstart-Thema hinzugefügt, das Ihnen den Einstieg in Amazon DocumentDB erleichtert. AWS CloudFormation	11. Januar 2019

Öffentliche Veröffentlichung	Dies ist die erste öffentliche Version von Amazon DocumentDB (mit MongoDB-Kompatibilität). Diese Version enthält den Entwicklerleitfaden und die integrierte API-Referenz für das Ressourcenmanagement.	9. Januar 2019

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.